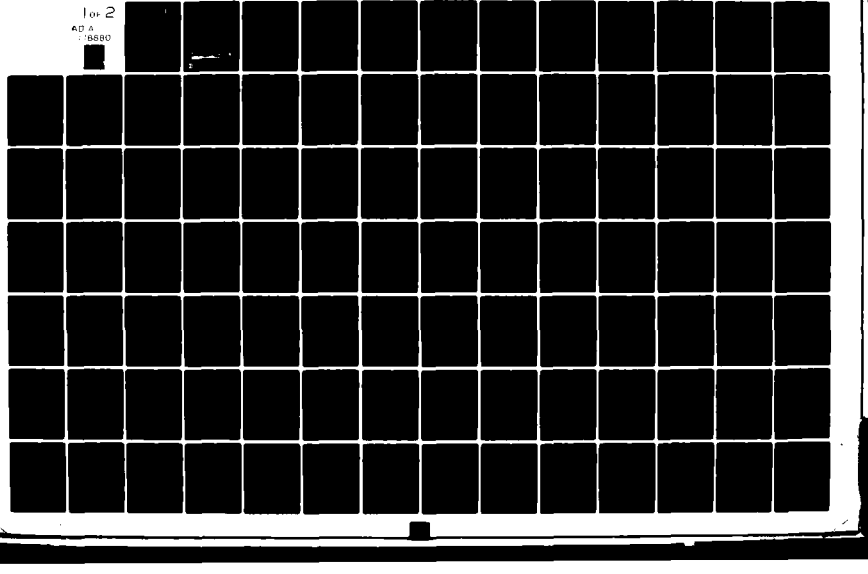
AD-A118 880 HARRIS CORP MELBOURNE FL GOVERNMENT ELECTRONIC SYSTE--ETC F/G 17/2
CODING FOR NAVY UHF SATELLITE COMMUNICATIONS (PHASE I).(U)
APR 77 N00039-76-C-0384
UNCLASSIFIED JA-6866 NL



PHOTOGRAPH THIS SHEET

AD A118850

DTIC ACCESSION NUMBER

II

LEVEL

I

INVENTORY

Coding for Navy UHF Satellite Communications (Phase I) Final Rept., 1 Apr. 77

DOCUMENT IDENTIFICATION

JA-6866

Contract N00039-76-C-0384

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DISTRIBUTION STATEMENT

ACCESSION FOR	
NTIS	GRA&I ☒
DTIC	TAB ☐
UNANNOUNCED	☐
JUSTIFICATION	
BY	Per Ltr. (FL-88, #82-1612, dtd. 28 July 82) on file
DISTRIBUTION / AVAILABILITY CODES	
DIST	AVAIL AND/OR SPECIAL
A	

DISTRIBUTION STAMP

DTIC COPY INSPECTED 2

DTIC
ELECTE
SEP 2 1982
S
D

DATE ACCESSIONED

82 08 02 050

DATE RECEIVED IN DTIC

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDA-2

DOCUMENT PROCESSING SHEET

DTIC FORM OCT 79 70A

AD A118280

JA 6866
N00039-76-C-0384

FINAL REPORT

Coding for Navy UHF Satellite Communications (Phase I)

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

HARRIS
COMMUNICATIONS AND INFORMATION HANDLING

HARRIS CORPORATION Electronic Systems Division
P O Box 37, Melbourne, Florida 32901 305/727-4000

JA 6866
N00039-76-C-0384

FINAL REPORT

Coding for Navy UHF Satellite Communications (Phase I)

For
Department of the Navy
Naval Electronic Systems Command
Washington, D.C. 20360

77 Apr 01

89652

EXECUTIVE SUMMARY

This report presents the results of the initial phase of a two-phase program to evaluate and verify the advantages associated with the use of error-correction coding in Navy UHF satellite communications, and ultimately to identify and recommend operational hardware for various Navy applications. In this first phase the problem was formulated, various techniques evaluated and the most appropriate one selected, and the performance of this technique predicted. In the subsequent phase a prototype unit will be constructed and through various test programs the usefulness of coding techniques in Navy satellite communication applications will be verified and specific hardware for operational deployment will be recommended.

The major technical problems in Navy UHF SATCOM systems are as follows:

- Insufficient available signal power, especially in aircraft and submarine links owing to the lack of significant antenna gain which can be provided on these platforms.
- Radio frequency interference (RFI), chiefly that arising from own-ship's communication and radar equipment.
- Signal level fading due to propagation effects, primarily ionospheric scintillation and multipath.

The first of these problems is of immediate concern in links to and from aircraft and submarines. Link power budgets for the FLTSATCOM system currently project sizable negative margins of E_b/N_o (-2.3 to -5.3 dB) for the forward TSCIXS channel and for both the forward and return SSIXS channels at 2400 bps. This means that under the conditions assumed in the power budgeting, adequate performance cannot be achieved at the required data rates on these links. While positive margins are predicted on links to and from surface ships using the AN/WSC-3 transceiver and OE-82B antenna, the margins are not generally large enough to support data rate increases to 4800 or 9600 bps which may be desirable in the future.

Thus, the signal power problem, while of near term concern in connection with aircraft and submarine communications, will be more generally felt after the deployment of FLTSATCOM as rates are expanded to more fully utilize the capacity of these satellites. Since the use of coding enables the system designer to achieve a specified performance at a lower level of E_b/N_o than is required in an uncoded system, coding has the potential for solving or at least contributing to the solution of these link power problems.

A major source of shipboard RFI is own-ship's radar or radar from nearby ships. At the lower data rates (2400 bps or less) the longest radar pulses are only a fraction of a bit long and if properly blanked by the blanking circuitry these pulses would cause minimal or negligible effect on the communications signal. However, substantial degradations due to these narrow,high-power pulses have been measured. These degradations are attributable to the inability of the blanking circuit to completely remove the effect of the pulse. When higher data rates are used, especially 9600 bps and above, one radar pulse may override several transmitted data bits, leading to an unacceptably high error rate in the data, even if ideal blanking could be accomplished. Through controlled insertion of redundancy, a carefully designed coded system has the capability of determining the identity of the affected bits, thus providing protection against this effect of pulsed RFI. Other sources of RFI such as intermodulation products and harmonics can desensitize communication receivers. Since this results in a depressed level of E_b/N_o, the use of coding provides some protection against this effect. For the same reason, coding offers some protection against the slow, frequency-flat fading characteristic of ionospheric scintillation and multipath.

For the above reasons, coding is potentially attractive in Navy UHF SATCOM applications. We surveyed a broad class of coding techniques to identify a technique which provided the necessary level of coding gain and protection against RFI, and at the same time satisfied a number of other constraints including:

- Size, weight and power requirements consistent with platform constraints.
- Compatibility with the existing and expanding inventory of Navy UHF SATCOM equipment.

- Transparency to operators of the system.
- Operability in TDMA as well as non-TDMA systems.

The result of this survey is the recommendation of pseudorandomly interleaved convolutional codes with maximum likelihood (Viterbi) decoding. Viterbi decoding is a practical and efficient technique with a proven record in SATCOM applications. We show in this report that in the absence of RFI effects we can provide sufficient coding gain to overcome the projected negative link margins noted above for the SSIXS and TSCIXS channels of FLTSATCOM. Furthermore, the use of interleaving enables us to correct bursts of RFI-induced erasures as well as provide moderate additional gain against thermal noise. Our recommended approach is demonstrated to be superior to several others both on technical and on operational grounds.

Some specific recommendations for the coder-interleaver combination are as follows:

- Because of the broad range of data rates to be employed, future Navy systems will require code rates of 1/2, 2/3, and 3/4.
- Longer code constraint lengths (R=1/2,k=9; R=2/3,k=10; and R=3/4,k=11) than have previously been used are recommended because 1 to 2 dB of additional coding gain can be obtained in the presence of RFI (depending on code rate and amount of RFI).
- The longer code constraint lengths can be implemented with modest complexity because of the relatively low data rates (32 kbps and lower) of interest in Navy applications, and the recent increases in the sizes of available random access memories (RAM).
- Pseudorandom block interleaving should be used to provide maximum robustness against the wide range of RFI parameters and data rates of concern.

In some applications, the full suite of functions might not be required. For example, in the low-rate SSIXS and TSCIXS links RFI is not a problem and the interleaver will not be required, so that a simpler unit can be employed in these links. However, in higher rate communications to surface ships, RFI may be a major

problem and some coding gain also required in order to support the required data rate. In such an application the full technique may be required.

In order to verify these concepts and to help in identifying requirements for operational equipment, we suggest a design for a flexible full-duplex prototype unit to include encoder/decoder, interleaver/deinterleaver, associated synchronization circuitry, and interfaces to the AN/WSC-3 transceiver-modem. We propose to implement such a unit and to use it in an extensive test program in order to verify the predicted performance of our recommended technique. These tests will be carried out using the shipboard RFI simulator recently constructed at the Naval Postgraduate School. Based on the results of these tests and on further study of operational requirements in various scenarios we will be in a position to recommend specific coding equipment required in various applications.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

1.0 INTRODUCTION

Harris Corporation, Electronic Systems Division, is pleased to submit this final report on Contract No. N00039-76-C-0384, the Navy UHF SATCOM Coding Study. Our efforts on this program can be categorized into the following major areas:

- Thorough study of the Navy UHF SATCOM environment, both technical and operational, and a realistic assessment of what coding can contribute to solution of the problems.
- A comprehensive survey of applicable coding techniques and selection of the technique which is most appropriate for this application.
- Detailed study of the selected technique including choice of codes, decoder and interleaver parameters and prediction of performance in the Navy SATCOM environment.
- Design of a prototype test model of the selected coder and decoder which can be used in simulation and in actual operational testing to prove the concepts and verify performance.

This final report is organized accordingly. In Section 2 we review the technical features of the UHF SATCOM channel and the constraints imposed by certain non-technical considerations. We argue that coding is an economical and effective means for enhancing the operational capacity of current- and future-generation Navy UHF SATCOM links, a means which can be integrated into existing equipments as well as employed in future designs. Based on our assessment of the problem and the potential for coding, we formulate a goal for selection of a coding technique to satisfy the Navy UHF SATCOM requirement.

Section 3 discusses our survey of possible coding techniques. We address the general choices and tradeoffs, consider in detail some candidates we studied and rejected, and explain our reasons for recommending interleaved convolutional codes with Viterbi decoding. A detailed study of the performance of this technique is contained in Section 4. We employ analysis, union

bounds, and computer simulations to predict performance under conditions modeling the actual environment. These results lead to choices of interleaver types, codes and decoder parameters.

Our recommended prototype design is discussed, at a top level, in Section 5. Our goal here is simply to show how the specific techniques identified in Section 4 will be realized in hardware. The details of our hardware design are discussed in our concurrently submitted Technical Proposal for Phase II of the program. Additionally in Section 5, we indicate the applicability of each of the functional elements of the prototype unit to various specific applications.

Finally in Section 6 we summarize our conclusions on Phase I and identify remaining technical questions which we feel should be addressed by actual hardware experimentation in Phase II.

The appendices contain several mathematical developments supplemental to the main text.

2.0 NAVY UHF SATCOM PROBLEMS AND CODING

Navy operational requirements dictate the need for reliable worldwide communication between various command centers ashore and fleet units at sea. Historically, the Navy has relied on HF circuits to fulfill this requirement for long-range communication with ships underway. However, HF communications have at least two disadvantages with profound strategic and tactical impact: the questionable dependability of the links due to varying propagation conditions, and the vulnerability to detection and direction-finding by hostile forces. The expanding technology of satellite communications offers a vehicle for worldwide communications which avoids or at least substantially reduces these two difficulties.

Many SATCOM users, notably common carriers and to some extent the other military services, have the option to configure their systems in such a way that a large number of geographically proximate low-capacity subscribers are serviced by a single high-capacity SATCOM terminal. In such a configuration, where the ratio of number of subscribers to number of terminals is high, elaborate and expensive terminals can be justified. However, since fleet units commonly cruise independently thousands of miles from friendly ports or stations, it is essential that each ship capable of independent operations have its own complete satellite communication terminal. Thus, the Navy SATCOM terminal inventory must number at least in the several hundreds, and probably in excess of 1000 even though each terminal will be of relatively low capacity. Under this constraint the cost of terminal equipment has a very substantial impact on the selection of a communication system architecture. Primarily because of this consideration, use of the military UHF band (240-400 MHz) has emerged as the most attractive option for the bulk of Navy satellite communication traffic. A requirement exists for Navy SHF terminals in order to ensure communications with other DOD users through the DSCS, and special-purpose Navy uses of K-band systems are likely to develop. However, UHF satellite systems such as the currently operational GAPFILLER and the projected FLTSATCOM and GP-SCS programs will provide the Navy's primary SATCOM capabilities for some time.

2.1 UHF SATCOM Technical Disadvantages

Although the economic arguments in favor of UHF satellite communication systems are strong, there are a number of technical problems in designing reliable UHF SATCOM links (some of which are not unique to the UHF band). The purpose of this study is to determine the extent to which coding can be used to help alleviate these problems. In the next few paragraphs we discuss at a summary level what these problems are and what contributions coding can make toward their solutions.

2.1.1 Available E_b/N_o

Efficient use of link power is an important consideration in the design of any satellite communication system, regardless of the frequency band. It is especially critical in a Navy UHF system because both economic and platform constraints dictate the use of transmitting and receiving equipment of modest capability, and because it is undesirable or impossible to provide significant amounts of antenna gain on some platforms, such as aircraft and submarines.

In an uncoded system, the available signal power at the receiver directly governs the data rate which the link can support reliably. When coding is used, the ratio of received energy per information bit to noise spectral density (E_b/N_o) required to achieve a given performance can be reduced, often by as much as 5 dB in practice. Thus, the signal power required to sustain a given data rate can be reduced significantly. This reduction in required received power level is termed "coding gain."

In order to be specific about the signal power problem in Navy UHF satellite communications, consider the plans for employment of the FLTSATCOM satellites. Navy accesses on these satellites are ten 25-kHz channels, one of which is a dedicated Fleet Satellite Broadcast (FSB) channel, with the other nine serving as two-way "tactical" channels. The intended initial usage of these channels is summarized in Table 2.1.1-1. All the tactical channels will initially be employed as 2400 bps channels

Table 2.1.1-1.

FLTSATCOM Channel Assignments

Channel	Usage	Data Rate
1	FSB	1200 bps
2	Secure Voice (AUTOSEVOCOM Interface)	2400 bps
3	CUDIXS	2400 bps
4	SSIXS	2400 bps
5	TADIXS	2400 bps
6	TSCIXS	2400 bps
7	Secure Voice (HICOM)	2400 bps
8	Secure Voice (FLT COMMON)	2400 bps
9	CUDIXS	2400 bps
10	TACINTEL	2400 bps

whose traffic is either data, as in the case of the Information eXchange Subsystems (IXS), or digitized secure voice.

Because of the special nature of the platforms served by Channels 4 and 6, these channels are allocated an additional 2 dB of EIRP from the satellite. Channel 4 is dedicated to the Submarine-Satellite IXS, which provides a SATCOM link to SSNs and SSBNs. Channel 6, which serves the Tactical Support Center IXS, provides communications between Tactical Support Centers and P-3 aircraft.

Net link margins from selected forward and return links through various FLTSATCOM channels are summarized in Table 2.1.1-2. These margins, excerpted from the FLTSATCOM system specification,[2-1] reflect the excess or deficiency of available power to achieve the required error rate performance ($P_b=10^{-5}$ for the data channels, $P_b=10^{-3}$ for secure voice and the FSB channel). It should be noted that the budgets leading to these link margins include 6 dB fade margin on both uplink and downlink and that the required values of E_b/N_o assumed for the specified error rate are higher than the theoretical values (apparently reflecting some provision for implementation loss), so that these margins may be somewhat pessimistic for normal conditions. Nonetheless, the projection of negative margin for both forward and report-back links of the SSIXS channel and the forward TSCIXS channel is a matter of some concern.

As we noted above, it is common for coding to provide an effective 5 dB of gain in satellite communication links. This gain would convert the negative SSIXS margins to positive and come within 0.3 dB of closing the TSCIXS forward link. From operational considerations, the use of coding is potentially more attractive in these two cases than the alternative of providing 3-6 dB of antenna gain on these platforms.

Although the IXS and secure voice subsystems are nominally designed for 2400 bps operation, all are expandable to 4800 bps or 9600 bps, and the 25-kHz repeater bandwidth can certainly support such rates without degradation. It is likely that the FSB data rate will increase, perhaps to as much as 9600 bps. Table 2.1.1-3 is a modification of the margins of Table 2.1.1-2, assuming all

Table 2.1.1-2.

Selected FLTSATCOM Link Margins

Channel	Platform	Rate	Margin, dB	
			Forward	Return
1	Ship	1200	3.3	NA
3	Small Ship	2400	6.4	10.5
4	Submarine	2400	-3.5	-2.3
6	P-3	2400	-5.3	3.4
8	Large Ship	2400	9.4	6.3
9	Large Ship	2400	5.9	5.0

data rates go to 9600 bps, all else remaining the same. These values indicate that links which are healthy at 2400 bps need additional power (or the equivalent) at 9600 bps. The use of coding to achieve some effective power gain can be a significant help in closing these links at the higher data rates. The desirability of coding at the higher data rates has been recognized in at least two current procurements of Navy SATCOM equipment. The Demand-Assigned TDMA Subscriber Unit[2-2] currently under development, which operates in a TDMA mode at 19.2 kbps BPSK and has a maximum symbol rate of 32 kbps QPSK, has an internal encoder and decoder. The General Purpose Modem[2-3] being developed to interface with the AN/ARC-143B, AN/WSC-3, and other UHF radios has special provisions for interfacing efficiently with decoding equipment.

2.1.2 Radio Frequency Interference

One of the major obstructions to successful shipboard satellite communications in the UHF band is interference, primarily that generated aboard the receiving ship. The problem arises because of the proximity of so many high-power emitters, primarily communications and radar, and its difficulty is compounded because of the complex electromagnetic environments surrounding the ship. A comprehensive investigation of shipboard RFI in the UHF band and its effects on communications has recently been carried out by Ohlson[2-4] and his associates at the Naval Postgraduate School. The results of their study have been, and will continue to be, of great value in the design of coding techniques, as well as other measures to improve shipboard satellite communications.

Ohlson, et al., discovered several major sources of interference which are potentially disruptive in the UHF band, including:

- Harmonics of HF transmitters
- Intermodulation products of HF and UHF transmitters
- Keying transients
- UHF radar transmitters
- Shipboard machinery and vehicles

Table 2.1.1-3.

Modified FLTSATCOM Link Margins

Channel	Platform	Rate	Margin, dB Forward	Return
1	Ship	9600	-5.7	NA
3	Small Ship	9600	0.4	4.5
4	Submarine	9600	-9.5	-8.3
6	P-3	9600	-11.3	-2.6
8	Large Ship	9600	3.4	0.3
9	Large Ship	9600	-0.1	-1.0

The character of the interference mechanisms which were observed can be classified as follows:

- Continuous (harmonics and IM products)
- Periodically pulsed (radar)
- Randomly occurring pulsed or impulsive (keying transients and machinery noise).

Ohlson, et al., suggest, quite correctly, the use of coding for protection against the latter two forms of interference. A major effect of the continuous variety of interference is receiver desensitization[2-5,2-6]. Since this has the effect of reducing E_b/N_o available at the demodulator, coding can, in principle, provide some protection against these interferers, since any coding gain helps to offset this reduction in E_b/N_o. However, it is difficult to accurately and realistically assess the level of protection coding may provide against CW interferers, and we believe that this issue should be left to experimental determination in Phase II of this program. In the case of the pulsed and impulsive interferers, however, coding methods can be sensibly designed and accurately evaluated, and in this study we have restricted our attention in the case of RFI to these forms of interference.

Table 2.1.2-1 exhibits the effects which the worst-case pulsed interferer has on data transmitted at standard rates from 75 bps to 32 kbps. The table indicates the duration of a pulse and the period of repetition, both measured in number of data symbols affected. The duty cycle of this interferer is seen to be approximately 5%. (For equivalent data on all shipboard pulsed interferers which are troublesome to UHF SATCOM, see Ohlson and Landry[2-4].)

All UHF SATCOM radios currently in use contain pulse-banking circuits which detect abrupt increases in input signal level and gate the demodulator input off until the signal level is restored to a nominal value. Pulses from own-ship's radar are normally strong enough to trigger the blanking circuit, so that effect of these pulses is simply to turn off the demodulator

Table 2.1.2-1.

Duration and Period of Worst-case Pulsed RFI

Data Rate, bps	Pulse Duration, bits	Period, bits
75	0.02	0.3
1200	0.24	5.2
2400	0.48	10.4
4800	0.96	20.9
9600	1.92	41.7
16000	3.20	69.6
19200	3.84	83.5
32000	6.40	139.1

for the duration of the pulse. (In practice, this blanking circuit will gate the signal off for somewhat longer than the pulse width, depending upon pulse width, power, IF bandwidth, and where in the IF chain the blanker is located; this pulse extension effectively increases the pulse widths indicated in Table 2.1.2-1.)

The table indicates that at the data rates currently in use (75-2400 bps), the most serious effect is the erasure of about one-half a symbol every ten symbols. With ideal blanking, both signal and noise are blanked for one-half a baud, and the degradation in overall bit error rate (BER) performance is not serious. In fact, however, Carlson and Ohlson[2-17] have observed substantial degradation with narrow high-power pulses, with and without blanking. Moreover, at the higher rates planned for future modems, interference pulses can totally erase several transmitted symbols, leading to a completely unacceptable error rate level. Appropriate use of coding will allow these erasures to be corrected (at the expense of some reduction in the information rate for a given symbol rate), thus restoring performance to an acceptably low BER. Depending on the strength of the code and the severity of the RFI, it is likely that in addition to repairing the RFI-induced erasures, some coding gain may be provided. Thus the coded system may achieve a specified BER at lower E_b/N_o in the presence of RFI than the uncoded system requires in the absence of RFI to achieve the same BER.

With proper pulse blanking, RFI from radar induces bursts of symbol erasures occurring periodically, with the net duty cycle governed by the radar duty cycle and the pulse extension (if any) of the blanking circuit. In principle, a coding scheme designed to make use of the periodicity (and hence, predictability) of the radar pulses, can outperform one which does not use this periodicity. However, there are many different values of symbol rate and radar PRF, and therefore many different RFI periodicities to be accommodated. Generally speaking, coding schemes designed specifically for one RFI period can degrade substantially if the period changes, even though the fraction of symbol erasures may not increase. Furthermore, even under conditions of fixed PRF and symbol rate, additional RFI arising from a second radar transmitter or from such aperiodic

sources as shipboard machinery or keying transients can degrade performance substantially. An essential requirement of a coding technique selected to combat shipboard RFI is robustness against such deviations from nominal operation. We placed great emphasis on meeting this requirement in our selection of a coding technique, since we believe that it is operationally impossible to control the shipboard RFI environment well enough to reliably and consistently predict the patterns of erasures.

2.1.3 Propagation Effects

UHF satellite communications are subject to two propagation phenomena whose effects are manifest in received signal strength variations: ionospheric scintillation and multipath. While the physical mechanisms in both cases are complex and difficult to precisely characterize, there are simple models which predict results in reasonable agreement with actual observations. Moreover, extensive experimentation and measurement programs have been undertaken using both experimental and operational UHF satellites to determine the effects of these two mechanisms. From the simple models and from the available test results we can glean enough information to be able to realistically assess how much protection coding can offer against these disturbances.

During the TACSATCOM Program, tests carried out by Army[2-7], Navy[2-8], and Air Force[2-9] agencies revealed that in the scenarios of interest in our problem, the most troublesome form of multipath arises when a strong specular reflected signal adds destructively to the direct signal to yield a net loss in received signal power. Some characteristics of this form of multipath, as determined by these tests, are:

- Significant fading occurs only at low elevation angles (up to 5^0 for shipboard terminals and up to 15^0 for aircraft terminals).
- Typical fade depths over water are 4-5 dB, with less severe fading observed over land.

- Fade rates on aircraft links are typically less than 1 Hz, except perhaps during take-off, landing and other maneuvers, when the rate may increase to a few Hz.
- To provide protection against multipath fading by frequency diversity requires a frequency separation on the order of 100 kHz for aircraft links, and much greater separation in the shipboard case. This implies that over the relatively narrow bandwidths of interest to us (up to 25 kHz) the fading can be considered frequency-flat.

These results are generally corrobated by other studies[2-10, 2-11]. Based on this data we can model the multipath effect as one of slow, frequency-flat fading, generally to depths of 5 dB but with occasional deeper fades.

The other fading mechanism of concern at UHF is ionospheric scintillation, which is caused by a variation in electron density in the ionosphere. Theoretical and experimental investigations of this phenomenon have been carried out for many years. Several of these are summarized in a Naval Air Development Center report[2-10]. Recent investigations of particular interest to us are the studies and experiments carried out by the Naval Electronics Laboratory Center[2-12] and Lincoln Laboratory[2-13, 2-14] using the Pacific GAPFILLER satellite. Generally, the conclusions of these studies are that:

- Scintillation is confined to zones near the geomagnetic poles and geomagnetic equator. It generally occurs at night and is a function, to some extent, of the season of the year and of the sunspot cycle.
- Fade depths are generally deeper than those occasioned by multipath, with 10 dB fades common during scintillation and even 25 dB fades not uncommon.
- Typical fade rates are less than 1 Hz with long fades of several seconds duration occurring commonly.
- Coherent bandwidth is of the order of several tens of MHz.

Thus, except for the fact that fades are typically deeper, scintillation fading is similar from a phenomenological point of view to that due to multipath. Both can be modeled as slow (in comparison with the data rate) and frequency-flat.

Based on these characteristics of multipath and scintillation fading, it is our judgment that coding ought not be viewed as a potential means for repairing their effects in all Navy UHF satellite communication systems. The major considerations which lead us to this conclusion are the following:

- Occurrence of these effects is restricted geographically (near the geomagnetic equator and poles for scintillation and at positions with low elevation angles to the satellite for multipath). Thus, these disturbances lack the universal character of the Gaussian noise and RFI discussed earlier.
- The fades which occur are typically several seconds long. In order to protect against such long fades, extremely long encoder memory is required and long decoding delays which would be objectionable in some kinds of traffic would be necessary.

Because of these considerations, we are inclined against taking steps in the design of the coding system specifically intended to repair the effects of these fades. However, we emphasize that because the fades are slow and frequency non-selective, any coding gain provided can be thought of as enhanced fade margin. This is a more modest, but more realistic, concept of what coding can do to help against these two phenomena. Furthermore, we reached our conclusion because we are attempting to evaluate coding for the general Navy UHF SATCOM problem; we recognize that there are special cases in which coding can and should be used specifically to combat multipath or scintillation. For example, the Naval Air Development Center report[2-10] cited earlier recommends the use of coding and increased antenna gain to reduce multipath problems in P-3 links through FLTSATCOM; and in conjunction with the Lincoln Laboratory scintillation program[2-14] extensive interleaving is being used with coding to overcome the effects of scintillation. It is important to recognize that both these programs are concerned with

relatively low rate (1200 to 2400 bps), one-way data transmission, so that effective code memory can be of reasonable size and long decoding delays are of no concern. Use of coding in these special cases is not at variance with our conclusion.

2.2 Operational and System-Level Constraints

We have maintained, and will prove subsequently, that coding has the potential for improving the capacity of links, protecting against shipboard RFI, and providing enhanced fade margins. To be a cost-effective approach, however, any technique nominated must, in addition to meeting the technical requirements, satisfy several other constraints. It is the purpose of this section to review some of these important constraints and determine their impact on the choice of coding techniques.

2.2.1 Operational Transparency

It is at least extremely desirable, if not absolutely necessary, that the employment of coding in a communications system have no impact that is apparent to users of the system (other than substantially improved system performance). One aspect of this transparency is that the physical requirements of the coding equipment must be modest. Size of equipment, for example, must be such that at most a small additional volume must be accommodated in the already crowded shipboard communications spaces. Likewise, equipment weight is an important physical constraint in an aircraft installation. An ideal solution would be to achieve size, weight, power requirements and form factors which would allow the equipment to be installed in empty spaces in existing cabinets, such as the ON-143 Interconnecting Group. In this regard, however, it must be noted that the encoder and decoder should be placed immediately outside the modem, and accordingly must be located in BLACK sections of the cabinets. (Most of the available space in the ON-143 is in the RED section.)

Another aspect of operational transparency requires that long encoding and decoding delays be avoided in voice links and

perhaps in some data links. This requirement was identified earlier as one of the major reasons why we rejected the use of coding as a general antidote to scintillation and multipath fading. These fades are typically several seconds in duration and the effective decoding delay would have to be several times this amount. This is clearly unacceptable in two-way voice communications, although it would not be detectable in a one-way transmission such as the Fleet Broadcast.

2.2.2 TDMA Operation

While some of the prime potential applications for coding in the near term are low-capacity, single access links such as SSIXS and TSCIXS, TDMA will become a widespread means of increasing system capacity in future Navy SATCOM links. Thus, the technique selected must be capable of operating properly within a TDMA hierarchy. This requirement has a number of implications for the coding technique.

TDMA systems commonly operate with a fixed "burst" or symbol rate. Redundancy required by coding is accommodated by a reduction in the input data rate in order to preserve the burst rate. In the interest of maximizing information capacity, it is desirable to use as high a code rate (ratio of number of information symbols to number of transmitted symbols) as the link margin will permit, even though the potential coding gain may be reduced. In high-capacity systems, if sufficient power is available to sustain the BER performance, protection against RFI pulses may be the only motivation for the use of coding and high rate codes are especially attractive in this case.

Since TDMA transmissions are organized into frames, it is desirable for the encoder output to also be partitionable this way in a natural fashion. This argues against the use of techniques requiring long terminating sequences which increase overhead. It is also desirable to avoid techniques which require variable amounts of decoding time, so that decoding can be accomplished on a frame-by-frame basis without additional buffering requirements.

Certain advantages also accrue to coding systems operating in TDMA links. One of the most significant is that because the TDMA system itself requires frame synchronization, some levels of encoder/decoder synchronization may be derivable from TDMA synchronization rather than being independently established, as would be required in a single-access link. It is desirable to consider all synchronization requirements at the outset of design, and to design to satisfy many of these requirements simultaneously, if possible.

2.2.3 Operability with Existing Equipments

Because of the large and increasing Navy inventory of UHF SATCOM equipment, a coding technique can only be cost-effective if it can be interfaced in an economical manner with equipment of existing design. It is highly desirable if this interfacing can be accomplished by an applique to the existing hardware, without requirements for any internal modifications.

Unfortunately, this is a constraint which is unlikely to be met in a manner entirely satisfactory to the designer of the coding equipment. The most efficient decoding techniques employ not simply hard symbol decisions from a modem, but some measure of likelihood. In BPSK and QPSK modems this measure can be obtained by quantizing the integrate-and-dump outputs more finely normally to 3 bits. Modems not intended to interface with decoders simply do not provide this kind of information; use of hard modem decisions costs approximately 2 dB in achievable coding gain.

A less widely appreciated consideration in retrofitting coding equipment to existing modems is that such modems are usually specified and designed for operation at a low symbol error rate (say, $P_E \approx 10^{-5}$) at a corresponding value of E_b/N_o. Well-designed modems may approach within a dB or so of the theoretically obtainable performance at this error rate, but depart substantially from theory at lower levels of E_b/N_o, for example, because of failure of the carrier loop to maintain lock. However, when coding is used the modem itself operates at a significantly reduced level

of signal-to-noise ratio in the symbol rate bandwidth, because of the combined effects of coding gain and expanded symbol rate bandwidth. For example, an uncoded PSK modem theoretically achieves $P_E=10^{-5}$ at 9.6 dB signal-to-noise ratio in the bit-rate bandwidth. If 5 dB of coding gain is achieved with rate-1/2 coding, the modem must operate at only 9.6-5-3=1.6 dB signal-to-noise ratio in the symbol rate bandwidth. Inferior modem performance at this depressed signal-to-noise ratio will degrade the available coding gain.

Because these and other modem considerations have such a significant impact on performance, and because modem modifications may be very expensive, these interfacing problems represent a major cost-effectiveness trade-off in selecting a coding scheme for widespread use.

The most common piece of UHF SATCOM equipment with which any proposed coding equipment will be interfaced is the AN/WSC-3 transceiver/modem. Recent test data obtained by Carlson and Ohlson[2-16] indicate that the actual BER performance of the AN/WSC-3 tracks within a fairly constant tolerance of the theoretical curve, even at high error rates, so that the problem of excessive degradation at low modem SNR apparently does not occur. Although designed as a hard decision modem, this unit has an available test point where an analog voltage corresponding to the integrate-and-dump output is supplied; however, the AGC at this point is inadequate to maintain the necessary level stability for operating with soft decisions. Instead, we intend to access the I&D input voltage and perform the I&D operation ourselves using the AN/WSC-3 clock signal for timing control. (Further details of this interface are given in our concurrently submitted Technical Proposal for Phase II.) At this level of investigation, it appears that for test and development purposes an adequate AN/WSC-3 interface can be established entirely external to this unit. For production equipment, however, it may not be satisfactory to rely on interfacing at this point, and we may be faced with the choice of employing a hard decision modem or making some hardware modifications

to the AN/WSC-3.* The same choice will be required for interfacing to other hard-decision modems such as the OM-43A and AN/SSR-1.

2.3 The Goal for a Coding Technique

With the foregoing technical characteristics and side constraints as background, we are in a position to formulate a realistic goal for a coding technique. We list below a number of features we would like the selected technique to exhibit:

- We are interested generally in high rate codes, between rates 1/2 and 3/4. Rates of 2/3 and 3/4 are primarily of interest in high-capacity applications such as TDMA, where the available satellite bandwidth is a constraint, while the rate -1/2 codes are of interest in lower-rate applications where maximum coding gain is the paramount consideration.
- The selected technique should be capable of providing substantial coding gain (4 to 5 dB) in the absence of RFI. This is important in the submarine and aircraft links where poor link margin, rather than RFI, is the dominant problem.
- Moderate coding gain (on the order of 2 dB) should be provided in the presence of RFI of up to 10% duty cycle. In view of the data listed in Table 2.1.2-1, 10% appears to be an overdesign. However, since a 5% duty cycle radar can conceivably by accompanied by other randomly occurring pulsed or impulsive interferers, or perhaps by another 5% duty cycle radar from an accompanying ship, this RFI level is a realistic worst-case design goal.
- The coding scheme must be robust against all RFI periodicities at data rates up to 32 kbps.
- Operability with existing and projected Navy SATCOM modems is required. Great advantage is attached to the capability of interfacing without modifications to the existing hardware.
- The hardware must have demonstrable legacy to a production version whose size, weight, and power characteristics are consistent with the platform limitations.

*A third alternative is to use the soft-decision General Purpose Modem[2-3] currently under development, interfacing with the AN/WSC-3 at IF.

- Coding delays sufficient to protect against radar interference are acceptable, but delays which would cause undue disruption of two-way voice communications are not acceptable.
- The coding scheme must be compatible with TDMA usage but also capable of single access usage. That is, no feature of the coding scheme must be inconsistent with the data framing characteristic of TDMA systems; yet we should not rely on external synchronization but be capable of providing the required encoder/decoder synchronization.

These are the characteristics which a coding scheme must possess in order to be a realistic and cost-effective solution to the Navy UHF SATCOM problem. Briefly stated, the remainder of this report discusses how we carried out a search for such a coding technique, and the performance and the means of implementation of the technique we have selected.

2.4 References

2-1. FLTSATCOM System Specification (U) FSCS-201, Naval Electronics Laboratory Center, 15 June 1976 (CONFIDENTIAL).

2-2. System Specification for a Demand-Assigned Multiple Access System, TT-A2-2206-0029, 19 March 1976.

2-3. Contract Specification, Modem, Digital Data, MD-( )(V)/U ELEX-M-240, 31 December 1975, with addendum of 1 May 1976.

2-4. J. E. Ohlson and T. C. Landry, "Shipboard RFI in UHF Satellite Communications" (U), Naval Postgraduate School Technical Report NPS-620L76103, October 1976 (CONFIDENTIAL).

2-5. R. F. Carlson and J. E. Ohlson, "Receiver Desensitization of the AN/WSC-3 Satellite Communications Set" (U), Naval Postgraduate School Technical Report NPS-620L76091, September 1976 (CONFIDENTIAL).

2-6. R. F. Carlson and J. E. Ohlson, "Receiver Desensitization of the AN/SSR-1 Satellite Communications Receiver" (U), Naval Postgraduate School Technical Report NPS-620L76101, October 1976 (CONFIDENTIAL).

2-7. "TACSATCOM Final Test Report" (U), U. S. Army Satellite Communications Agency, 1 December 1970 (CONFIDENTIAL).

2-8. "Navy Tactical Satellite Communication (TACSATCOM) Technical Test Report" (U), Naval Electronic Systems Command, 1 September 1970 (CONFIDENTIAL).

2-9. "USAF TACSATCOM Test Program Final Report" (U), Air Force Systems Command, Electronic Systems Division, 1 February 1971 (CONFIDENTIAL).

2-10. W. Schoppe, "Characteristics of Multipath in an Aircraft-Satellite Communication System" (U), Naval Air Development Center, 9 September 1974 (CONFIDENTIAL).

2-11. I. L. Lebow, K. L. Jordan, and P. R. Drouilhet, Jr., "Satellite Communications to Mobile Platforms," Proc. IEEE, Vol. 59, pp. 139-159, February 1971.

2-12. R. U. F. Hopkins and M. R. Paulson, "Equatorial Scintillation Space Diversity Experiment," Proc. National Telecomm. Conf., Vol. III, paper 43.3, November 1976.

2-13. E. A. Bucher, "UHF Satellite Communication During Scintillation," Technical Note 1975-10, Lincoln Laboratory, MIT, August 1975.

2-14. E. A. Bucher and D. P. White, "Time Diversity Modulation for UHF Satellite Communication During Scintillation," Proc. National Telecomm. Conf., Vol. III, Paper 43.4, November 1976.

2-15. Technical Manual, Satellite Communication Set AN/WSC-3, NAVELEX 0967-LP-453-7010.

2-16. R. F. Carlson and J. E. Ohlson, "Bit Error Rate Measurements on the AN/WSC-3 and AN/SSR-1 Satellite Communication Sets" (U), Naval Postgraduate School Technical Report NPS-620L76102, October 1976.

2-17. R. F. Carlson and J. E. Ohlson, "Performance of the AN/SSR-1 Satellite Communications Units in the Presence of Pulsed Radar Noise" (U), Naval Postgraduate School Technical Report NPS-620L76122, December 1974 (CONFIDENTIAL).

3.0 CODING TECHNIQUES SURVEY

A major task of this study was to consider applicable coding techniques in the light of the requirements and constraints unique to the Navy UHF SATCOM problem and to select the technique best suited to this application. This section details our survey of coding techniques and shows why we selected interleaved convolutional codes with Viterbi decoding. First we discuss some general aspects of coding technique selection. Next we consider in some detail three candidate techniques which we felt had significant promise but rejected in favor of the selected technique. Finally, we discuss why we made the ultimate selection we did.

Our primary concern here is with the specific properties of various coding techniques in the light of the goals established in Paragraph 2.3. Additional background material on these coding techniques and on their performance potential in other applications can be found, if desired, in one of the standard references on coding theory [3-1,3-2,3-3].

3.1 General Considerations and Tradeoffs

The requirement to protect against bursts of errors or erasures induced by pulsed RFI implies that codes and decoding algorithms capable of burst correction be selected. Classically, there are two approaches to the burst correction problem. One is the use of codes whose structure is specifically suited to the correction of error patterns occurring in bursts. The other approach is to interleave the encoder output sequence prior to transmission and to deinterleave following reception so that errors are distributed more uniformly at the input to the decoder. With the later technique a code with a specific burst correction capability is no longer required and a conventional random error correcting code is used.

It can be shown that for a fixed fraction of symbols in error (or erased), and for similar encoder and decoder complexity, somewhat greater efficiency can be achieved by the use of burst

correction coding than by the use of interleaved random error correction coding, provided that the code parameters are well-matched to the burst characteristics. Thus in cases in which the burst characteristics are well understood, burst correction codes are generally to be preferred. However, as we noted in Paragraph 2.1.2, the technique we select must be robust against considerable variation in burst parameters, a characteristic which is not often exhibited in ad hoc burst correction schemes. Therefore the use of interleaved random error correction coding techniques is suggested. (However, the interleaving can also be susceptible to degradation in the presence of varying RFI parameters and the interleaving technique must be selected carefully for robustness as will be discussed subsequently.)

Aside from the issue of robustness, even when the burst can be corrected properly burst correcting codes do not normally provide much protection against background errors, so that our goal of providing modest coding gain in the presence of RFI would be in jeopardy with this choice.

Another reason for our inclination in favor of interleaved random error correcting codes is flexibility. In some important applications such as the 2400 bps SSIXS and TSCIXS links, RFI is not a major problem and the primary notivation for the use of coding is to provide coding gain. By omitting the interleaver/deleaver and using only the coder decoder, the same equipment is easily adaptable to this kind of application.

The choice of block codes or convolutional codes represents another design tradeoff. In a block code, as the name implies, successive blocks of data symbols are encoded independently into blocks of symbols for transmission, and received blocks are decoded independently back into blocks of data symbols. In a convolutional code, a data sequence continuously drives a linear sequential circuit (the encoder) so that the transmitted sequence is the convolution of the impulse response of the encoder with this data sequence.

Each of these classes of codes has its own group of distinguished proponents to argue the merits of their choice.

In practice, convolutional codes have tended to be preferred in satellite communication applications while block codes have been widely used on other channel types such as telephone lines. In addition to channel properties, such details as the desired data rate and the hardware state-of-the-art influence the selection. Convolutional coding equipment operating up to several Mbps is commercially available from a number of vendors. It is likely, however, that for extremely high-speed applications (several tens of Mbps), sophisticated block coding techniques will be preferable to convolutional coding.

Although there is a precise theoretical sense in which convolutional codes are superior to block codes of equivalent complexity [3-4], the major practical attraction of these codes is that efficient optimum or near-optimum decoding algorithms exist when they are used on memoryless channels. These algorithms are the maximum likelihood decoding technique introduced by Viterbi[3-4] and the suboptimum technique of sequential decoding initially proposed by Wozencraft [3-5] and subsequently refined by Fano [3-6]. With both of these techniques the decoder can make use of symbol likelihood information produced by a soft decision modem, thus improving performance substantially over that available using only hard symbol decisions. In contrast to these schemes, practical decoding techniques for block codes have been suboptimum and primarily oriented toward the use of hard decisions. As a result, generally longer and more complex block codes are required to furnish performance competitive with that available from shorter and simpler convolutional codes.

We have established a goal to provide subtantial coding gain in the absence of RFI and moderate gain even with 10% symbol erasures. At the present time such gains are most practically achieved through the use of convolutional codes and soft decision modems, at least at the data rates of interest to us. One apparent disadvantage of convolutional codes is the lack of an obvious "blocking" or "framing" of encoder output data in order to be compatible with a TDMA system. However, the output of a convolutional encoder can be blocked by periodically inserting a constraint length of zeros, and for short codes and reasonably long frames the increase in overhead is quite small.

The foregoing remarks indicate our inclination toward the use of interleaved convolutional codes for the Navy UHF SATCOM application. At the first level of investigation, these codes seem to provide the best match to the goals set forth in Paragraph 2.3. However, we did study in considerable detail the performance of some important representatives of other techniques as well as this one. In the next section we will show how the results of these investigations confirmed our choice.

3.2 Some Specific Candidates

In the previous section we identified two dichotomies: the choice of block versus convolutional codes, and the choice of burst correction versus interleaved random error correction. Figure 3.2-1 illustrates this classification of coding techniques and cites several specific techniques we investigated in detail.

Among the class of random error correcting block codes, the Bose-Chaudhuri-Hocquenghem (BCH) codes [3-7,3-8] are the best known and among the most powerful. We considered interleaved BCH codes and found that they fell short of our established goals. Even in the absence of thermal noise, interleaved BCH codes cannot protect against 10% erasures with code rates in the range of interest. These results are summarized in Paragraph 3.2.1.

The Reed-Solomon [3-9] codes are a powerful class of burst correcting block codes. We investigated several variants of Reed-Solomon techniques, but as we show in Paragraph 3.2.2 we concluded that no high-rate techniques could provide sufficient coding gain in the presence of RFI. While a number of burst correcting convolutional codes are known [3-2], we felt that none of these techniques would outperform the Reed-Solomon codes and thus did not investigate them in detail.

In the class of interleaved convolutional coding techniques, we considered the use of sequential decoding [3-5,3-6]. This technique is capable of providing very large amounts of coding gain in the absence of RFI and substantial gain in the presence of RFI, but as we show in Paragraph 3.2.3, it has a number of disadvantages which led us to reject its use in this application.

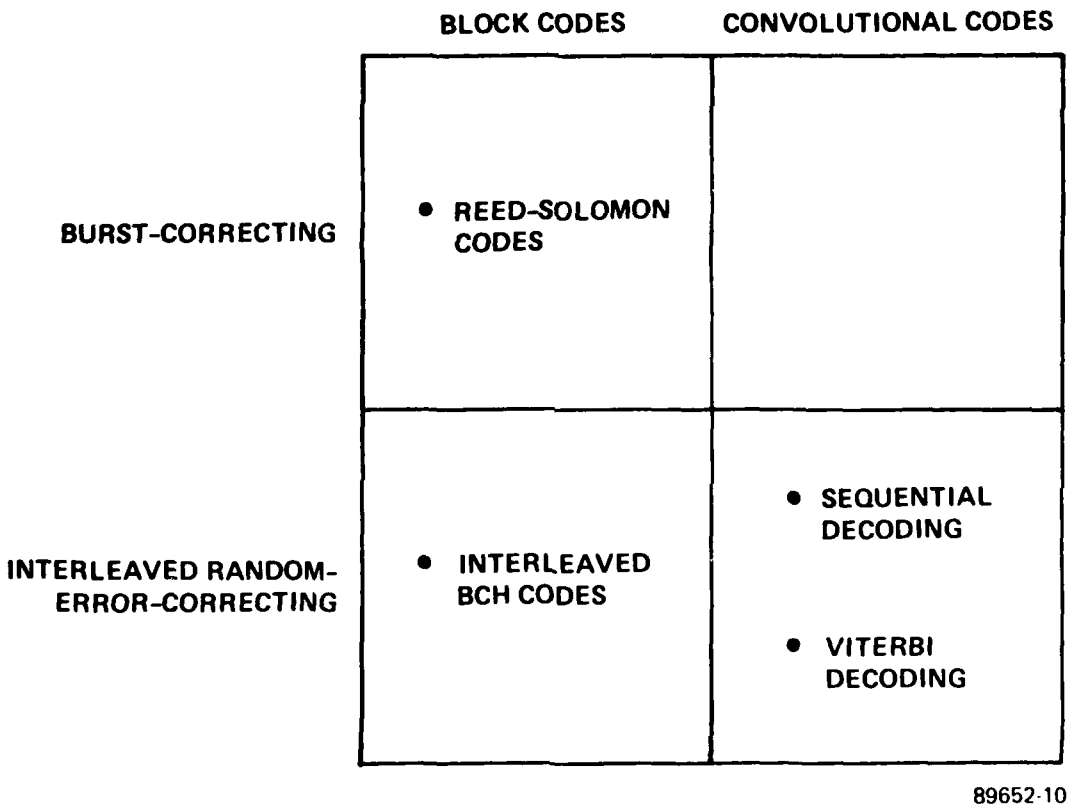


Figure 3.2-1. Classification and Examples of Coding Techniques

Finally, we have found in the use of Viterbi decoding [3-4] a technique which is consistent with all the goals established in Paragraph 2.3. Our reasons for selecting it are summarized in Paragraph 3.3, and details of the recommended technique are discussed in subsequent sections of the report.

3.2.1 Interleaved BCH Codes

BCH codes comprise a large class of good binary random-error-correcting block codes. In our application, BCH codes could be used in conjunction with random interleaving to disperse RFI-induced erasures. However, as we shall show, this approach does not provide adequate performance. In these arguments we will use without proof, several properties of BCH codes. For the details of these proofs the reader is referred to Peterson and Weldon [3-2].

Specific codes will be denoted by the parameters (n, k) where n refers to the code block length and k refers to the number of data bits per block. Then the number of parity bits per block is n-k and the code rate is R=k/n. Generally, the most efficient codes are the primitive BCH codes with block length $n=2^m-1$. The number of parity bits required for these codes to guarantee minimum distance d=2t+1 is approximately

$$n - k \approx mt \qquad (3.2.1\text{-}1)$$

By adding one overall parity check the distance can be increased to d=2t+2. A code with minimum distance d can correct up to d-1 erasures caused by RFI. Thus, in order to correct a fraction of erased bits equal to δ, the code distance must satisfy

$$d - 1 > \delta n$$

or

$$t > \frac{\delta n - 1}{2} \qquad (3.2.1\text{-}2)$$

From (3.2.1-1), the code rate is given approximately by

$$R \approx 1 - tm/n \qquad (3.2.1\text{-}3)$$

By combining (3.2.1-2) and (3.2.1-3), we obtain an approximate bound on the maximum rate which can be used to correct a fraction δ of erasures. This is given by

$$R \approx 1 - \frac{(\delta n-1)m}{2n} \qquad (3.2.1\text{-}4)$$

For δ=0.1, this bound is tabulated in Table 3.2.1-1. Note that as n increases, the maximum usable code rate decreases.

With randomly occurring erasures, the number of erased bits per codeword is binomially distributed with mean δn. There will be many cases with more than δn erasures per codeword, so one cannot realistically expect good performance near the rates given by (3.2.1-4) unless n is very large (so that with high probability the number of erasures is near δn). The probability of a decoded bit error at an erasure rate δ can be estimated from

$$P_b \approx \sum_{i=d}^{n} \frac{i}{2n} \binom{n}{i} \delta^i (1-\delta)^{n-i} \; . \qquad (3.2.1\text{-}5)$$

This equation assumes no thermal noise-erasures are the only source of errors-and thus provides an optimistic estimate of performance. Using (3.2.1-5), we computed P_b for all BCH codes with R>0.56 and $n \leq 127$ with δ=0.1. In all cases $P_b \geq 10^{-3}$. In addition, we note from Table 3.2.1-1, that there is no way of achieving a high code rate with δ=0.1 and for n>127. Thus, we conclude that binary BCH codes cannot even provide the necessary level of RFI protection with high code rates, let alone provide any coding gain in the presence of RFI. This approach is therefore not suitable for our application.

3.2.2 Reed-Solomon Techniques

The Reed-Solomon codes form an important class of non-binary BCH codes which are particularly attractive in burst correction applications.

Table 3.2.1-1. Maximum BCH Code Rates for Correction of 10% Erasures

m	n	R
4	15	.93
5	31	.83
6	63	.75
7	127	.68
8	255	.62
9	511	.56

Symbols in Reed-Solomon (RS) codes are elements of the field $GF(2^m)$, the Galois field with 2^m elements. Since elements of this field can be represented as binary m-tuples, each RS symbol can be identified with m binary symbols (bits). This is illustrated in Figure 3.2.2-1, which shows how a non-binary (n,k) RS encoder operates with a binary data source and a binary modem. A block of mk data bits from the data source is input to the encoder, which regards these as a block of k symbols over $GF(2^m)$. A linear operation is performed on these symbols (linear, that is, with respect to the arithmetic of $GF(2^m)$), to produce a block of n output symbols over $GF(2^m)$. It is a characteristic of RS codes that the block length in symbols is given by $n=2^m-1$. These n output symbols, each represented as a binary m-tuple, are handed to the modem for transmission as mn transmitted bits. The structure of an RS code word is further illustrated in Figure 3.2.2-2. External to the encoder and decoder, the (n,k) non-binary RS code can be viewed simply as a (mn,mk) binary block code. Inside the encoder and decoder however, arithmetic is done in $GF(2^m)$, and it is <u>symbol</u> errors rather than transmitted <u>bit</u> errors which the decoder attempts to correct.

These codes have a natural capability to correct short bursts of errors or erasures of transmitted bits. Note that any burst of m+1 or fewer consecutive transmission errors can cause at most two symbol errors. (A symbol error occurs if at least one of the corresponding m transmitted bits is in error.) Thus, an RS code with minimum distance 5 is capable of correcting such a burst error.

As we noted above, the block length of RS codes is $n=2^m-1$ symbols. For any k, the minimum distance between RS code words (in symbols) is given by

$$d = n-k+1 \qquad (3.2.2-1)$$

It can be shown that this is the greatest achievable minimum distance for any (n,k) linear code, so that in this sense, RS codes are optimum. To get an idea of the implications of this optimality

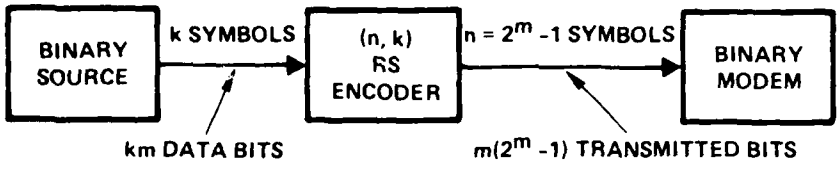


Figure 3.2.2-1. Reed-Solomon Encoding

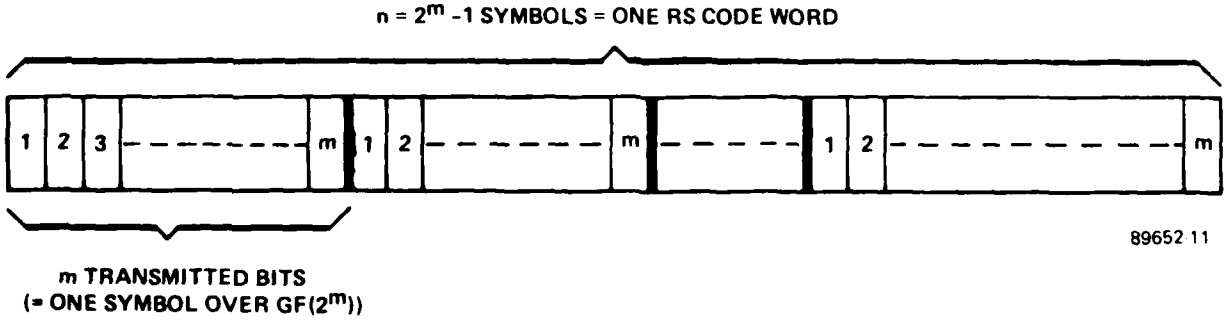


Figure 3.2.2-2. Structure of a Reed-Solomon Codeword

in our problem, suppose that the communication of the transmitted bits is perturbed by periodic erasure bursts but no other disturbances. Suppose that the burst mechanism is such that exactly δn symbols of every n-symbol RS codeword are erased (where δn is assumed to be an integer). Any pattern of δn symbol erasures can be corrected if the code minimum distance is at least $\delta n+1$. Thus, from (3.2.2-1) we require

$$n-k \geq \delta n$$

or

$$1-\frac{k}{n} \geq \delta$$

Since k/n is the code rate R, we are capable of correcting all the erasures if

$$R \leq 1-\delta$$

But the capacity of the channel whose only disturbance is the erasure of a fraction δ of the transmissions is $C=1-\delta$. Thus, with RS codes, we can operate at channel capacity for this particular erasure mechanism. However, this result obtains only because the erasure pattern is precisely fitted to the code structure, such that the channel erasure pulse is exactly an integer number of RS symbols and is aligned such that the leading edge coincides with the start of a symbol. Any deviation from this alignment can lead to substantial degradation. For example, if an erasure is exactly one symbol long, but is not precisely aligned with one of the symbols, its effect will be to erase two symbols in the codeword rather than one. Thus the effective erasure probability with which the decoder must contend is actually twice the erasure probability on the raw channel. Similar erasure amplification effects occur as the erasure mechanism departs from the above ideal case in other respects. The amount of amplification depends on the transmission rate and the RFI parameters. It is these erasure amplification effects which place the primary limitations on RS code performance in our application. (In addition to correction of erasures, of course, we would like to be able to correct random symbol errors occurring due to Gaussian noise.)

In addition to the conventional RS approach, we investigated two alternative techniques based on RS coding. The first is to randomly interleave the RS encoder output symbols prior to transmission, as shown in Figure 3.2.2-3. The motivation for this approach is to more uniformly balance the number of symbol erasures which must be corrected in each code word. In the absence of such interleaving some code words can have considerably more symbol erasures than others. With interleaving, the tendency is to reduce the incidence of substantial departures from the average number of symbol erasures in each codeword and provide robustness against changes in data rates and RFI periods.

In the other alternative technique an overall parity check is added to the m-bit sequence representing each RS encoder output symbol, and the resulting sequence of bits is randomly interleaved prior to transmission. Thus, the scheme can be characterized as a concatenation of the RS code with a distance-2 (m+1,m) inner code, with random interleaving of transmitted bits. This is diagrammed in Figure 3.2.2-4. Erasure bursts on the raw channel are dispersed by deinterleaving so that the inner code works against random bit errors and erasures. Since its minimum distance is 2, it can correct single bit erasures, thus reducing the number of symbol erasures handed to the RS decoder. If more than one bit erasure appears in an inner codeword, the corresponding RS symbol is erased. If hard decisions are made by the demodulator, the inner code can detect single errors and erase the corresponding RS symbol. However, the added redundancy due to the inner code also provides a means to improve performance if the demodulator makes soft decisions.

Performance of all three of these RS techniques is examined in detail in Appendix D. Performance depends on transmission rate and RFI parameters, since these quantities govern the amount of erasure amplification. At a nominal transmission rate of 20 kbps and with RFI arising from one or two sources similar to the worst-case source of Table 2.1.2-1 (i.e., a 5% or 10% raw channel erasure rate), our conclusions on RS code performance are as follows:

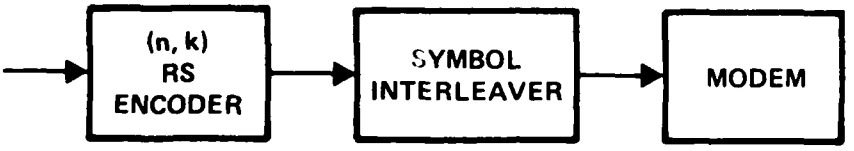


Figure 3.2.2-3. Interleaved RS Encoding

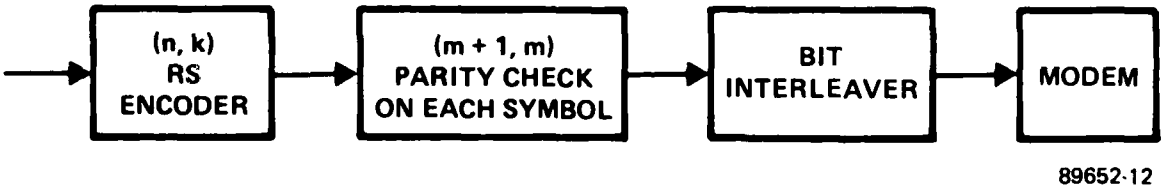


Figure 3.2.2-4. Concatenated RS and Parity-Check Encoding with Bit Interleaving

- RS codes with no interleaving-Moderate coding gain can be achieved for R≈2/3 with one RFI source. When two RFI sources are present, no gain can be achieved for $R \geq 2/3$, but a moderate gain can be achieved at $R \approx 1/2$. This technique is sensitive to changes in data rate.
- RS codes with random symbol interleaving-Moderate gain can be achieved at $R \approx 2/3$ for one RFI source, but no gain is available for two sources at $R \geq 1/2$. This technique is robust to changes in data rate.
- Concatenated codes with random bit interleaving-With two RFI sources, there is no gain available for $R \geq 2/3$, but there is a small amount of gain available at $R = 1/2$ and possibly a large amount available at lower code rates. This technique is robust to changes in data rates.

Our desired performance level is a coding scheme that offers a moderate coding gain (around 2 dB) and corrects erasures caused by pulsed RFI with duty cycle 0.10 and can operate at code rates up to 3/4. The Reed-Solomon codes can meet these requirements at code rates $R \leq 1/2$ but not at the higher rates. Thus, this technique is not suitable for our application.

3.2.3 Sequential Decoding

Sequential decoding is a technique for decoding convolutional codes which can provide substantial coding gain on sufficiently well-behaved noisy channels, such as the deep-space channel and certain satellite communication channels. The amount of computation a sequential decoder must perform per decoded bit is a random variable, since the difficulty of decoding depends on the channel noise (in contrast with most other techniques, in which the decoding computation per bit is fixed regardless of the channel noise). It is this variability of computation, and the consequent need for high decoder speed (with respect to the data rate) and large buffer storage, which are the pracitcal determinants of decoder performance. Sequential decoders are normally operated with codes so powerful that the probability of a decoding error is negligible.

In the Navy UHF SATCOM problem, we are concerned with computational performance of sequential decoders operating on the

additive white Gaussian noise (AWGN) channel with some of the transmitted symbols erased by the blanking circuit. Sequential decoders are known to be extremely sensitive to any clustering of noise or other disturbances, so we will assume here that the symbol erasures occur independently at random throughout the data. This is a reasonable model, provided that an interleaving strategy is employed such that the deinterleaving disperses the erasures randomly. For example, the interleaver and deinterleaver may consist of large blocks of storage with data read in sequentially and read out in a pseudo random pattern. If the block size (interleaver period) is large enough and if there are no peculiarities arising from commensurate interleaver period and RFI period, the random erasure model is valid.

The amount of computation a sequential decoder must perform per decoded bit is a random variable with a distribution of the form

$$\Pr\{C>N\} \approx KN^{-\rho} \quad , \; N\to\infty \qquad (3.2.3\text{-}1)$$

Such a distribution is called a Pareto distribution. The Pareto behavior occurs if the sequential decoder makes no decoding errors and if the data sequence is arbitrarily long so that the extent of the decoder's search is not bounded. In a practical decoder, some limit on searching must be imposed, and as a result, (3.2.3-1) holds only for some limited range. Over the range of interest, however, both simulations and tests of real sequential decoders have verified the Pareto behavior in the tail ($N>>1$) of the distribution of computation.

The value of the Pareto exponent ρ is determined by the code rate and the channel statistics. If $\rho\leq 1$, the average value of C is infinite. In fact, the Pareto distribution has the property that the ρth and higher moments of C fail to exist. The theoretical minimum acceptable value of the exponent is $\rho=1$, and under certain circumstances, operation at ρ near 1, with generally large computation per decoded bit, is feasible. However, in two-way systems, excessive decoding delay must be avoided, and this argues for

operating with a larger value of the exponent. Experience and study of performance of commercially available units suggest a design value of $\rho=2$, although it must be stressed that this represents only a fairly conservative estimate, and that determination of the actual required ρ to achieve satisfactory performance is a complex problem, probably requiring simulation and/or hardware testing.

Appendix C discusses in detail the relationship among ρ, code rate, erasure rate, and E_b/N_o. Table 3.2.3-1, which is compiled from the results obtained in Appendix C, summarizes the required E_b/N_o to achieve $\rho=2$ for various rates, hard and soft modem decisions, and erasure rates of 0, 5%, and 10%. Note that in the absence of erasures, these results indicate that up to 9.6 - 5.0 = 4.6 dB of coding gain can be achieved in the absence of erasures with rate - 3/4 codes and soft-decision demodulation. With 10% erasures, the available gain is reduced to 9.6 - 7.2 = 2.4 dB, which is certainly in consonance with our goal of providing modest gain in the presence of 10% RFI. Since $\rho=2$ is a conservative design, the practically achievable coding gains are likely to be somewhat higher.

These results indicate that with respect to the goals established for coding gain and RFI protection, sequential decoding exhibits satisfactory performance. Unfortunately, there are a number of other practical considerations which discourage us from recommending this approach. Primary among these is that sequential decoding is extremely sensitive to any channel behavior which does not affect symbol transmissions independently. In view of the unpredictable character of the UHF shipboard channel, there is some doubt as to how closely the predicted gains can actually be approached. We believe that this doubt can only be resolved by testing of such equipment on an actual or accurately simulated channel. Consequently it is our opinion that while sequential decoding is potentially attractive from the point of view of performance, the technical risks associated with this technique are not consistent with the primary objective of this program.

Table 3.2.3-1.

Required E_b/N_o to Achieve $\rho=2$

(All Values in Decibels)

Code Rate	No Erasures		5% Erasures		10% Erasures	
	Hard Decisions	Soft Decisions	Hard Decisions	Soft Decisions	Hard Decisions	Soft Decisions
1/4	5.7	3.7	6.0	4.0	6.3	4.3
1/2	6.3	4.2	6.8	4.6	7.4	5.1
2/3	7.0	4.7	7.7	5.3	8.6	6.1
3/4	7.5	5.0	8.4	5.9	9.9	7.2
4/5	7.8	5.3	9.0	6.4	12.5	9.1

3.3 The Recommended Technique

The major result of our survey is the recommendation of interleaved convolutional codes with Viterbi decoding. The major factors leading to this selection are as follows:

- We demonstrated, and will support in Section 4.0, that primary technical goal of simultaneously providing modest coding gain and protection against 10% duty cycle RFI with high rate codes can be met.
- Viterbi decoding is a proven concept, widely applied in SATCOM systems and not requiring major technical and hardware breakthroughs. Technical risks associated with this technique are thus modest.
- We believe this technique will degrade more gracefully than others as the channel characteristics depart from the idealized models we have assumed.
- This technique is very flexible. It can be used with hard or soft modem decisions (although soft decisions are strongly preferred). In low rate applications the interleaver can be discarded, thus substantially reducing the system complexity. This technique is capable of deriving synchronization internally or using externally furnished synchronization. Efficient, relatively simple hardware can be built to operate at all data rates of interest.

In Section 4.0 we will explain the selected techniques in greater detail and provide extensive performance data to predict its performance in the Navy UHF SATCOM environment. In Section 5.0 we discuss further our proposed implementation and show how it can be configured in a variety of modes for different specific applications.

3.4 References

3-1. E. R. Berlekamp, Algebraic Coding Theory, McGraw-Hill, 1968.

3-2. W. W. Peterson and E. J. Weldon, Jr., Error-Correcting Codes, Second Edition, MIT Press, 1972.

3-3. S. Lin, An Introduction to Error-Correcting Codes, Prentice-Hall, 1970.

3-4. A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Trans. Inform. Theory, Vol. IT-13, pp. 260-269, April 1967.

3-5. J. M. Wozencraft, "Sequential Decoding for Reliable Communication," IRE Convention Record, 1957, Part 2, pp. 11-25.

3-6. R. M. Fano, "A Heuristic Discussion of Probabilistic Decoding," IEEE Trans. Inform. Theory, Vol. IT-9, pp, 64-74, April 1963.

3-7. R. C. Bose and D. K. Ray-Chaudhuri, "On a Class of Error-Correcting Binary Group Codes," Inform. and Control, Vol. 3, pp. 68-79, 1960.

3-8. A. Hocquenghem, "Codes Correcteurs d'Erreurs," Chiffres, Vol. 2, pp. 147-156, 1959.

3-9. I. S. Reed and G. Solomon, "Polynomial Codes Over Finite Fields," J. SIAM, Vol. 8, pp. 300-304, 1960.

4.0 DISCUSSION OF THE RECOMMENDED TECHNIQUE

For the reasons noted in Paragraph 3.3, we recommend the use of convolutional encoding - Viterbi decoding with pseudorandom interleaving for the higher data rate applications (say 4800 bps and higher) where pulsed RFI from radar is a serious problem. For the lower data rates (2400 bps and lower) where RFI affects only a portion of a bit, and in applications where RFI is not a problem at all, we recommend convolutional encoding - Viterbi decoding without interleaving.

A considerable effort has been devoted to determining the most efficient way to implement this approach, and this section addresses our conclusions as to the implementation and our assessment of achievable performance (i.e., coding gain and RFI protection) in the Navy UHF SATCOM environment. Paragraph 4.1 is concerned with the choice of interleaver type, its size, and the means of implementation. Next in Paragraph 4.2 we discuss the operation and implementation of convolutional encoders and maximum likelihood (Viterbi) decoders. Finally in Paragraph 4.2 we present results on performance of interleaved convolutional codes with Viterbi decoding in the presence of thermal noise and blanked RFI, and we demonstrate that this technique has the capability to provide the necessary coding gain and RFI protection identified in Section 2.

4.1 Interleaver Considerations

As shown in the shipboard RFI study by Ohlson and Landry[4-1], There are a variety of error sources in a shipboard environment and the environment is quite variable from ship to ship. This makes it extremely difficult to select a coding-interleaving approach to take advantage of the error characteristics to allow a more effective decoding job to be done, since an approach that may work well on one ship may work badly on another. As a consequence it is desirable to use an approach that will be subject to as little variations in performance as possible as the RFI environment is changed. In other words, our goal is to use an approach with maximum robustness. This

section will discuss our selection of interleaving approach, the choice of size, and the method of synchronizing the interleaver and deinterleaver.

4.1.1 Interleaving Approach

Interleaving is used to disperse bursts of erasures caused by blanked RFI pulses as well as errors or erasures caused by other RFI sources. The means of accomplishing this is to permute the encoder output sequence prior to modulation, and to perform the inverse permutation on demodulator outputs before decoding. Thus the decoder input data is in the same sequence as the corresponding encoder output, but the bursts of erasures are broken up by the deinterleaver.

The most troublesome form of interference is pulsed periodic RFI arising from radars. For a given transmitted bit rate, let P denote the RFI period in transmitted bits. If δ is the effective duty cycle (including pulse extension in the blanker, if any), then the maximum degree of erased bit separation is achieved if the deinterleaver produces an erasure once every K bits, where K alternates between $\lfloor 1/\delta \rfloor$ and $\lceil 1/\delta \rceil$ in such a way that the average interval between erasures is $1/\delta$. For a fixed P and δ, it is possible to build an interleaver and deinterleaver which achieves this maximum level of separation between erasures. This approach is well-known as periodic interleaving. However, for different values of P and δ, the structure of the interleaver will be completely different. The period P is governed by both the radar PRF and the transmission rate, and we must operate over a broad range of both these parameters. Moreover, in addition to predictable radar-induced RFI, we are interested in protecting against other kinds of pulsed and impulsive interferers whose occurrence cannot be predicted as precisely. For these reasons we rejected the above periodic interleaving strategies in favor of a strategy which is more robust against variations in RFI characteristics.

Suppose first that it were possible to deinterleave randomly, in the sense that if the symbol erasure rate is δ at the input to the deinterleaver, then at the deinterleaver output each symbol has

probability δ of being an erasure, independent of all other symbols in the deinterleaver output sequence. Such a random deinterleaver is robust since the statistical character of the deinterleaver output depends only on the rate of channel symbol erasures, and not on how they are distributed. A disadvantage of randomly interleaving is that with a fraction δ of all bits erased the overall system performance exhibits a floor on the achievable error rate. If two codewords differ in d places, then there is a probability of δ^d that erasures will occur in those places, and thus probability $\frac{1}{2}\,\delta^d$ that at least one data bit error will be made by the decoder. Thus, even at infinite E_b/N_o, the coded system cannot achieve an error rate less than $\frac{1}{2}\,\delta^d$. In fact, the error rate floor is usually found to be an order of magnitude or more above $\frac{1}{2}\,\delta^{d_{min}}$, where d_{min} is the minimum distance of the code. An upper bound on the error rate floor is found by using the union bound (from Appendix A) i.e.,

$$P_b(\text{floor}) \leq \tfrac{1}{2} \sum_{i \geq d_{min}} w(i)\, \delta^i \,, \qquad (4.1.1\text{-}1)$$

where w(i) is the total information weight of all codewords of weight i for the code used. Because of this error rate floor, we see that random interleaving provides robustness at the sacrifice of performance at very large values of E_b/N_o.

The practical approximation to random interleaving is to use a pseudorandom interleaver and deinterleaver. One implementation of such a device is a large block of memory which is loaded sequentially and unloaded in a pseudorandom sequence. This interleaver, in fact, approaches the behavior of a random interleaver as the block length becomes infinite. From this viewpoint a long interleaver block length is desirable, but in a practical system a long block makes system complexity and delay increase. Thus, in considering these tradeoffs in a practical system the choice of interleaver length is a compromise. The resulting block size should retain most of the robustness of the purely random strategy, but there will be some sensitivity to the structure of the interference.

In some cases the performance of the pseudorandom approach will be better than that which would be obtained with a purely random strategy and in other cases it will be worse. This effect has been demonstrated by Richer[4-2] in simulations of Viterbi decoding with pseudorandom interleaving in the presence of periodic bursts of erasures. He found that the particular interleaver he was using (length of 256) resulted in decoded error rates smaller than those predicted for purely random interleaving by a factor of 1.5 to 2.0. Thus, we believe that, as was done in this case, the particular pseudorandom sequence should be tested against probable RFI periodicities to obtain the best match. However, since a real RFI environment may produce a large fraction of erasures and errors without well-defined periodicities we anticipate that the performance difference between a purely random interleaver and a given pseudorandom interleaver will narrow.

Once an interleaver size is selected one must find a good pseudorandom sequence for performing the interleaving. The desirable characteristics of the pseudorandom sequence include:

- Randomness
- Random distribution of periodic bursts of erasures
- Good performance with the selected code in the presence of periodic bursts of erasures
- Good performance with the selected code in the presence of actual shipboard RFI

These characteristics are presented in increasing order of difficulty of verification of performance.

Richer[4-2] has found interleaver structures satisfying the first three characteristics using a "linear congruential sequence" (see Knuth[4-3]). This approach produces a sequence of numbers satisfying the relationship

$$A_{n+1} = (aA_n + c) \bmod M. \qquad (4.1.1\text{-}2)$$

For a fixed M the sequence may be changed by changing a and c. To obtain a maximal length sequence (i.e., a sequence with period M)

the parameters a and c must satisfy the following conditions:

- a and c must be less than M
- c must be relatively prime to M
- (a-1) must be a multiple of p for every prime p dividing M
- (a-1) must be a multiple of 4 if M is a multiple of 4

This approach then generates a sequence of numbers from zero through M-1. The sequence will have the characteristic of randomness if a is relatively prime to M[4-3]. Since each admissible choice of a and c produces a different sequence, then with proper selection of interleaver block length M there will be many sequences which satisfy (4.1.1-2).

The sequences generated in this manner all have the "randomness" property, but the candidate sequences should be further tested to insure that they randomly distribute periodic bursts of erasures. This can be done rather easily, as Richer[4-2] has shown, by tabulating the spacing between erasures and the probability of successive erasures at the output of the deinterleaver for numerous hypothesized periodic interference structures. He found interleaver structures that resulted in deinterleaver output statistics that had somewhat greater separation between erasures than random for the interference structures of interest. In subsequent simulations using that interleaver structure, the performance was better than if a purely random interleaving strategy had been used. Thus, we believe an examination of the deinterleaver output statistics for hypothetical interference structures gives a good indication of how the deinterleaver will perform.

The final desirable characteristic we mention is good performance in the presence of actual shipboard RFI. This can only be verified through testing in a simulated or actual shipboard environment.

4.1.2 Interleaver Size

Choice of interleaver size is necessarily a compromise between several conflicting characteristics. The size should be:

- Large enough to provide "typical" deinterleaved sequences
- Large enough so that frame sync bits do not increase overhead too much
- Small enough that there is not excessive complexity and decoding delay
- Small enough to avoid excessive data loss during sync and resync

In insuring that the interleaver is large enough to insure "typical" output sequences, a principal consideration is that for a given periodic RFI structure producing a fraction δ of all bits as erasures, any given deinterleaver loading have its fraction of erased bits, δ^*, be nearly equal to δ. This can be achieved if the interleaver is long enough to contain many periods of RFI. Then the worst case loading is when there is one extra burst of erasures in the deinterleaver as illustrated in Figure 4.1.2-1. That is, there are $\rho+1$ bursts of erasures in ρ periods of interference. For this case $\delta^* \leq \frac{\rho+1}{\rho}\,\delta$. Using the worst-case RFI conditions from Table 2.1.2-1, we have about 139 bits in a period. For this period, a block length in the range 1000 to 2000 will cause only a small amount of nonuniformity in the fraction of erasures per deinterleaved block.

Another consideration is the amount of overhead allocated to frame synchronization. We have tentatively selected a 36 bit sync word. This word length allows the frame sync performance goals to be met. If this word were transmitted every 1000 bits, the sync overhead would be 3.6% while if it were transmitted every 2000 bits the sync overhead would be only 1.8%.

In considering the reasons to keep the interleaver size small one problem occurs at the lower data rates. For a fixed interleaver size the system delay introduced is inversely proportional to the data rate. The amount of delay introduced is equal to one block length at each end. There is apparently no need for interleaving for symbol rates less than 4800 bps. At 4800 bps, a 1000-bit block length introduces an additional delay of about 0.4 seconds end-to-end, and a 2000 bit block length gives a delay of 0.8 seconds. At 9600 kbps these numbers become 0.2 sec and 0.4 sec,

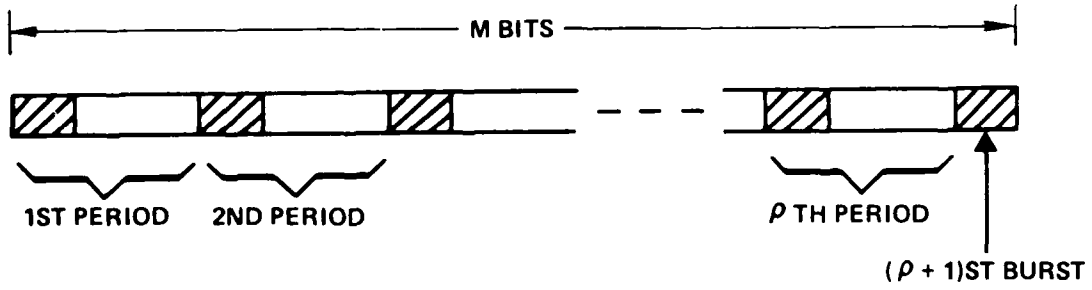


Figure 4.1.2-1. Worst-Case Deinterleaver Loading

respectively. It appears that block lengths above 2000 will introduce undesirable delays at the lower data rates.

The best compromise among these tradeoffs appears to be a block length in the neighborhood of 1000. It may also be desirable to impose additional constraints on the block size. For example, in a TDMA application, the block size might be chosen to divide the frame length, so that deinterleaver synchronization can be derived from TDMA frame synchronization, without separate synchronization circuitry and sync pattern overhead. Similarly, if the block length is chosen to be a multiple of the number of bits per code branch, then any required decoder synchronization can be derived from deinterleaver synchronization.

4.1.3 Interleaver-Deinterleaver Synchronization

In the most general case, some means of properly synchronizing the interleaver and deinterleaver must be provided. In what follows, we discuss and analyze a fairly common sync strategy, and indicate typical values of frame sync parameters. The synchronizer has two modes of operation. During initial acquisition it is in the SEARCH mode. Here the parameter tradeoffs involve the probability of missing the sync word (P_{miss}), versus the probability of false sync (P_{fs}). During normal operation the synchronizer is in the LOCK mode. Here the parameter tradeoffs involve the probability of recognizing an out-of-sync condition (P_{ros}) versus the probability of falsely thinking that sync has been lost (P_{drop}).

In the calculations that follow, we assume that the frame synchronizer uses hard decision data for simplicity of implementation. In addition, there is no requirement for the interleaver block length and the synchronizer frame length to be identical. One could make the frame length several blocks long. In our implementation we found it convenient to make the frame length equal to two interleaver blocks (or about 2000 bits). This has the advantage of reducing the frame sync overhead to 1.8% (for the 36 bit sync word).

When in the SEARCH mode, on the average, the synchronizer will look 1000 times before the correct sync word appears. If parameters are chosen such that $P_{fs} \leq 10^{-4}$ per look, then in 1000

looks the probability of false sync will still be only about 0.1. Thus, we will set $P_{fs}=10^{-4}$ as our goal. Letting the sync word length be denoted by m, then a threshold k will be established. That is, if in matching the m bits in the sync word with the true sync word there are k or fewer mismatches, then the sync word is accepted as correct. In this case the synchronizer then reverts to the LOCK mode.

The probability of false sync in one trial with random data is simply the probability that random data matches the correct sync word in all but k or fewer positions, i.e.,

$$P_{fs} = 2^{-m} \sum_{i=0}^{k} \binom{m}{i}. \qquad (4.1.3\text{-}1)$$

When the correct sync word (with channel errors) is compared with itself, the probability of miss is simply

$$P_{miss} = \sum_{i=k+1}^{m} \binom{m}{i} p^i (1-p)^{m-i} , \qquad (4.1.3\text{-}2)$$

where p is the channel error rate. The worst-case value of p that can be seen and still result in reasonable decoded error rate (10^{-3} or less) is in the neighborhood of 0.1 to 0.12 (this would be with a rate one-half code with E_b/N_o near 0 dB and an erasure rate of 0.1).

The parameters m and k are chosen to yield suitable values of P_{fs} and P_{miss}. As we noted, for a frame length of 1000, $P_{fs}=10^{-4}$ is a reasonable goal. To keep acquisition time reasonably short, it is desirable to achieve $P_{miss}\leq 0.2$.

Once the synchronizer has entered the LOCK mode, the synchronizer continues to look for the frame sync pattern in the expected positions. An "out-of-lock" condition is not declared until there are more than k' mismatches in a sync pattern comparison r consecutive times. Thus if a bit slip occurs, it will take at least r frames to recognize. The probability of an out of sync detection is

$$P_{ros} = (P_d)^r \qquad (4.1.3\text{-}3)$$

where

$$P_d = 2^{-m} \sum_{i=k'+1}^{m} \binom{m}{i}. \qquad (4.1.3\text{-}4)$$

is the probability of more than k' mismatches in random data. We want P_{ros} to be near one. On the other hand, we need to keep the probability of false dropping sync when we are actually in sync extremely small. Here we have

$$P_{drop} = (P_{fd})^r \qquad (4.1.3\text{-}5)$$

where

$$P_{fd} = \sum_{i=k'+1}^{m} \binom{m}{i} p^i (1-p)^{m-i} . \qquad (4.1.3\text{-}6)$$

is the probability of more than k' mismatches when properly synchronized.

Choice of the parameters k' and r is made to achive realistic design probabilities of correctly and falsely indicating "out-of-lock," as given by (4.1.3-3) and (4.1.3-5). It should be noted that it may not be possible to achieve desired values of P_{ros} and P_{drop} with the value of m selected earlier to achieve satisfactory performance in the SEARCH mode, and m may have to be increased.

Avoiding the details of the calculation, we have found that the following set of frame sync parameters yield reasonable performance at p=0.12:

Frame length = 2000 bits
m = 36
k = 6
k' = 12
r = 2

The resulting overhead is 1.8%. Synchronization performance achieved is

$$P_{fs} = 3.5 \times 10^{-5}$$
$$P_{miss} = 0.133$$
$$P_{ros} = 0.935$$
$$P_{drop} = 2.95 \times 10^{-8}$$

Average acquisition time can be estimated rather easily since P_{fs} and P_{miss} are small. The frame synchronizer will not begin the search until an entire frame is received. Then on the average it is one-half of a frame until the correct sync word is seen. Thus, if P_{miss} and P_{fs} were zero, the acquisition time would be 1.5 frames. However, if the correct sync word is missed it is one more frame before it is seen again and sync is achieved. In addition, in the one-half frame that is being searched before the correct sync word appears, false sync could occur with probability $\approx 1000\ P_{fs} = 0.035$. If this happens it will take two frames to recognize that the sync is false and one additional frame in the SEARCH mode to see the correct sync word again. The the average acquisition time in frames is *approximately*

$$T_{acq} \approx 1.5 + P_{miss} + 3\ (1000\ P_{fs})\ , \qquad (4.1.3\text{-}7)$$

which corresponds to approximately 3500 bits.

The calculations we have made have assumed that the sync word is corrupted by random bit errors. Because of the nonrandom nature of the RFI we feel that it is very important to pseudorandomly interleave the sync word so that synchronization cannot be affected by RFI-induced burst errors. It would be foolish to interleave the data to protect it and not do the same for sync bits, which are even more critical.

4.2 Decoder Considerations

The use of convolutional coding with maximum likelihood decoding has found wide application in communication systems because it has typically been the most practical technique for achieving

large power gains (on the order of 5 dB). In this section we will discuss encoder and decoder principles and implementation.

4.2.1 Convolutional Codes

We provide here a very brief discussion of convolutional codes. More detail as well as figures are provided in Appendix A. A convolutional encoder is a linear, finite state machine which accepts an input sequence of bits and produces an output sequence with controlled redundancy to allow error correction. A rate 1/2 encoder is shown in Figure A-1. (Rate 1/2 implies that two output bits are generated for each input bit).

Convolutional code structure has historically been illustrated by the code tree as shown in Figure A-2, but this representation masks the repetitive structure of convolutional codes. Two infinite information sequences which differ in only one position will be represented as two paths in the code tree. However, because the encoder is a finite state machine, the corresponding transmitted sequences can only differ over a span of k branches, where k is the constraint length of the code in branches (after k-1 identical information symbols are shifted into each encoder they must both be in the same state and will have identical outputs thereafter). The repetitive nature of convolutional codes is illustrated nicely by the trellis diagram shown in Figure A-3. Each node represents an encoder state. All nodes in the first row represent state 0, all nodes in the second row represent state 1, etc. Each segment in the trellis represents the transition from one state to another and the bits that are output from the decoder during that transition. Two paths are said to "diverge" when they both leave the same state at the same time, and two paths are said to "merge" if they both enter the same state at the same time. The convention used in Figure A-3 is that the upper branch leaving each node is due to an information bit 0 and the lower branch is due to an information bit 1.

Assume without loss of generality that the all zero path is the correct path. Then any path with a finite number of non-zero information bits diverges from the all zero path at some node and merges with the all zero path at some later node becoming identical

with it thereafter. For example, the path corresponding to a single information bit one preceded and followed by all zeros is ... 00 00 11 01 11 00 00 ... which is identical to the all zeros path for all but three branches. The merging property of convolutional codes is the basis for the Viterbi algorithm.

4.2.2 Maximum Likelihood Decoding Algorithm

Maximum likelihood decoding implies that the decoder chooses the sequence out of all possible transmitted sequences which correlates best with the received sequence. However, with the Viterbi algorithm the number of possible transmitted sequences that must be examined is limited by using the merging property of convolutional codes. That is, when two paths merge at a given node in the trellis, only the path with the larger correlation with the received sequence need be examined further. The reason for this is that the two paths will be identical from that point on since they are starting from the same state, and the path with the larger correlation at merging will always have the larger correlation further into the trellis. Thus, if every time two paths merge the path with the smaller correlation with the received sequence is discarded, then no path is discarded which has a chance of being the path with the largest correlation. Maximum likelihood decoding car then still be performed even though certain paths are discarded as described above.

The operation of a maximum likelihood decoder using the Viterbi algorithm is very straightforward. The decoder for an R=1/2, constraint length k, binary, convolutional code stores 2^{k-1} paths, one for each state, and the correlation of each path with the received sequence. Each time a branch is received the decoder extends the path stored in each state by forming both the upper and lower branch and computing the increment to the accumulated correlation for each of these branches. The decoder now has 2^k paths with two of them entering each state. Now for each state the accumulated correlations (or equivalently the path metrics) for the two paths in that state are compared and the smaller is discarded. This portion of the decoding process is called path metric updating and comparison or "add-compare-select" (ACS). Then the 2^{k-1} resulting paths and their

correlations with the received sequence are stored. This process is repeated when the next branch is received. Thus, the decoder will always have stored in it the path which will eventually be the highest correlated path. The completion of the decoding process requires that one of the stored sequences be selected for output. The best procedure is to select that stored path which has the largest correlation with the received sequence and output its oldest branch.

This is a brief description of the maximum likelihood decoding algorithm. Considerably more detail regarding implementation is presented in Appendix B including some discussion of hardware minimization. Topics addressed in this appendix include decoder branch synchronization, the "add-compare-select" function, storage and updating of information sequences, and the output decision device.

The most widely implemented code with maximum likelihood decoding is the R=1/2,k=7 code. Since the decoder complexity grows exponentially with k, there are practical reasons for not using large values of k. At high data rates k=7 is about as large as can be used while maintaining reasonable decoder complexity. Thus, this code has become quite commonly used for this reason. This decoder has 64 states. However, at low data rates where the states can be updated serially, the reasons for using k=7 are not so compelling. An increase in constraint length affects mainly the memory requirements. With large RAM's now widely available a R=1/2,k=9 decoder with 256 states can be implemented while maintaining a relatively small part count. The R=2/3,k=10 and R=3/4,k=11 also have 256 states and can be implemented with a decoder structure very similar to the R=1/2,k=9 decoder. The serial implementation which we envision will accommodate all data rates up to 32 kb/s which is sufficient for all forseeable information rates for Navy use. It will be shown in the next paragraph that the increase in code constraint length we are recommending provides about 1 dB of additional coding gain in the presence of RFI at all code rates.

4.3 Performance Predictions

In this section we will briefly describe the approaches we have used in estimating the performance of our selected technique and give the resulting performance curves for R=1/2(k=9), R=2/3(k=10), and R=3/4(k=11). These curves will present the performance assuming PSK signaling on an AWGN channel both with and without erasures due to RFI. For the purposes of this performance evaluation the RFI is modelled as randomly occurring erasures. This is a good, though not exact, model of the RFI. The reasons are that the shipboard environment is more complex than this and in a practical system one would have to implement pseudorandom rather than random interleaving. However, the resulting performance predictions provide a very good basis with which to make relative comparisons between several coding techniques.

4.3.1 Performance Estimation Techniques

There are two principal tools we have used in evaluating the performance of our proposed technique - union bounds and decoder simulations. The union bound gives an upper bound on decoded bit error rate that is very tight at low output error rates but loose at high output bit error rates. The simulations give an estimate of bit error rate, but it is subject to statistical fluctuation. This makes it particularly difficult to obtain accurate estimates of performance by simulation at very low error rates, say $P_b \approx 10^{-5}$. Thus, the simulations were performed at signal-to-noise ratios which give $P_b \geq 10^{-4}$. These two performance estimating approaches complement each other very nicely. The simulations give very accurate performance predictions at the higher values of P_b while the union bound gives accurate predictions at low values of P_b.

The technique used in calculating a union bound is discussed in detail in Appendix A. The equation used for calculating this bound for a R=(n-1)/n code is given by

$$P_b \leq \frac{1}{n-1} \sum_{j=0}^{\infty} w(j)\, P_j \qquad (4.3.1\text{-}1)$$

which is identical to equation (A-7) of Appendix A. The term w(j) is the total information weight for all sequences with path weight j for the code being evaluated. This term was determined by using a code weight structure tabulation program for determining all significant values of w(j) for any code. The term P_j is probability that any incorrect path that differs from the correct path in j positions will be selected by the decoder. This term is typically calculated using equation (A-9) of Appendix A which assumes no quantization of the demod output and unrestricted range in the stored path metrics. However, in order to achieve more accuracy in predicting the effects of 8-level quantization of the demod output and restricted range in the stored path metrics, we take a different approach. This is achieved by using equation (A-15) of Appendix A in calculating P_j. By doing this, predictions of decoder performance can be obtained accurate to within about 0.1 dB at $P_b=10^{-5}$.

The decoder simulation program we have used accurately reflects the implementation techniques we use including 8-level quantization, branch metric assignments, path metric updating technique, and a fixed decoding depth for making bit decisions. The simulations were typically run long enough to produce about 100 or more decoded bit errors to obtain a reasonably accurate statistical average. Of course, this limits the feasibility of using simulation to make error rate predictions at the very low error rates because of the long simulation time required. Therefore since we have achieved very close agreement between simulated results and union bound predictions we have relied mainly on the union bounds to predict performance at $P_b=10^{-5}$.

4.3.2 Performance of the Selected Technique

We see two types of environments in which the decoder might be used. First, on links with negative margin and no RFI problem (such as the SSIXS and TSCIXS channels of FLTSATCOM) the solution is to obtain as much gain as possible in Gaussian noise. Here we believe a R=1/2,k=9 code is indicated. The resulting coding gain is 5.6 dB. The other environment is on shipboard (either TDMA or non-TDMA) where the RFI environment is severe. Here we desire to correct RFI induced

errors and erasures and provide some gain against Gaussian noise. Since the data rates used in this application vary over a wide range and at the higher data rates one may need to reduce the code redundancy, we recommend evaluation of several code rates so the code rate may be tailored to suit the application. The choices offered are R=1/2(k=9), R=2/3(k=10), and R=3/4(k=11). In general, we recommend choosing the code rate that will maximize the transmitted data rate and meet the performance goal. Thus, if there is plenty of link margin one would tend to choose the highest rate code while if link margin is a problem one would select a lower code rate to increase coding gain.

Although R=1/2, 2/3, and 3/4 codes have been published in the literature, we did some code searching to improve performance and/or decrease implementation complexity. Larsen[4-4] published a R=1/2,k=9 code that achieved the upper bound on minimum free distance (d=12). We searched all k=9 codes with d=12 to find the one with best performance at high signal-to-noise ratio (i.e., the code with the minimum total information weight for all distance 12 paths). It turned out that the best code was the code found by Larsen. Paaske[4-5] has published a list of good R=2/3 and 3/4 codes. Here we chose to use other codes we have found that simplify the decoder implementation. The code generators for each of these codes is indicated subsequently in the figures showing the performance curves. No effort has been made to make the codes either transparent or non-transparent (see the discussion of the synchronizer in Appendix B for the implications of this). We simply have chosen the best code whether it be transparent or nontransparent. When transparent codes are used polarity reversals can be corrected either by the frame synchronizer or through the use of differential encoding/decoding. When nontransparent codes are used this can be done in either the frame synchronizer or the decoder branch synchronizer.

The performance of the R=1/2 code is given in Figure 4.3.2-1. Note the close agreement between simulated and union bound performance estimates. These estimates reflect <u>all decoder implementation degradation</u> for our implementation approach. This code gains about 5.6 dB

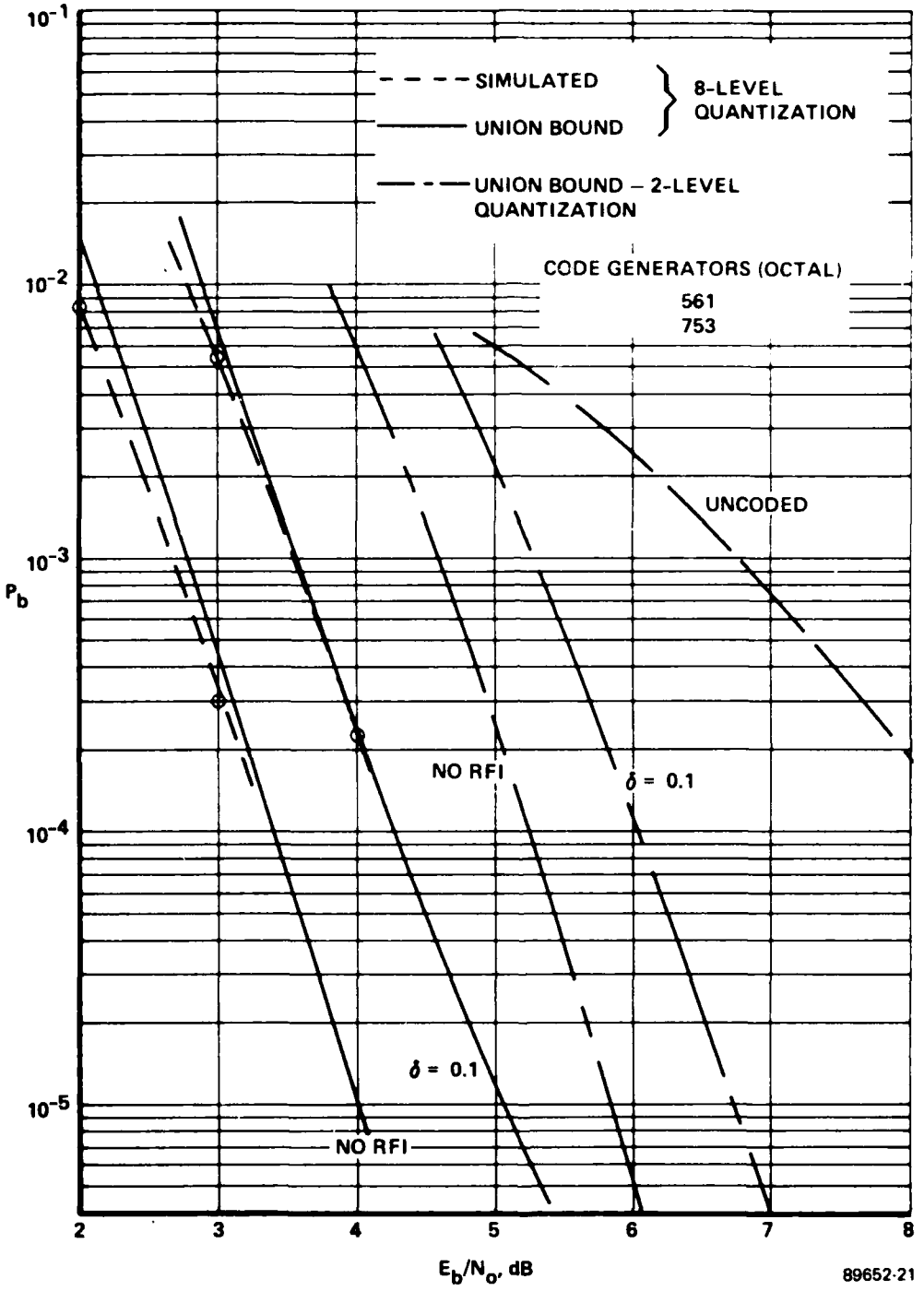


Figure 4.3.2-1. Performance of the R=1/2,k=9 Code

over uncoded at $P_b=10^{-5}$ when used with PSK on an AWGN channel. This compares with 5.0 dB of gain that is obtained by the R=1/2,k=7 code which is the most commonly implemented code to date. The rationale for going to the longer constraint length is that significant additional coding gain is obtained for all code rates both against Gaussian noise and severe RFI. The additional gain obtained ranges from 1 to 2 dB with R=2/3 and R=3/4 in the presence of RFI.

The figure also shows the performance when a fraction δ of the received bits are erased by RFI. This assumes the presence of a blanker to turn off the input to the demod when an RFI pulse is present. Of course, in the AN/WSC-3 the blanker does exactly this. In this case with 10% of all bits erased by RFI the decoder still provides 4.5 dB of coding gain.

The coding gain figures discussed above assume soft decisions or 8-level quantization of the demod output. If this is inconvenient to do, one can still provide coding gain if hard decisions or 2-level quantization is used. It is well-known that in doing this the performance will degrade by about 2.0 dB. To illustrate this we show in Figure 4.3.2-1 union bounds on performance when using hard decisions. In this case the performance degrades by about 1.8 dB. Thus, with no RFI as we indicate in the figure about 3.8 dB of gain can be obtained using hard decisions. We require absolutely no circuit modifications to the AN/WSC-3. However, the normally-employed differential encoding/decoding must be bypassed by positioning a strap. If RFI is present the interface is a little more difficult. We must use a blanking signal either from the AN/WSC-3 or from an external source to "erase" the bits affected by RFI at the decoder input. If this is done about 2.9 dB of gain is obtained with 10% of the bits affected by RFI. The degradation shown here when operating with hard decisions is typical and such curves will not be shown for R=2/3 and R=3/4 to avoid cluttering the bit error rate curves.

Figure 4.3.2-2 shows the performance of the R=3/4,k=9 code found by Paaske[4-5]. The simulation results were obtained at Lincoln Labs[4-6]. These simulations assumed no quantization of the demod output voltage. Assuming that one would actually implement

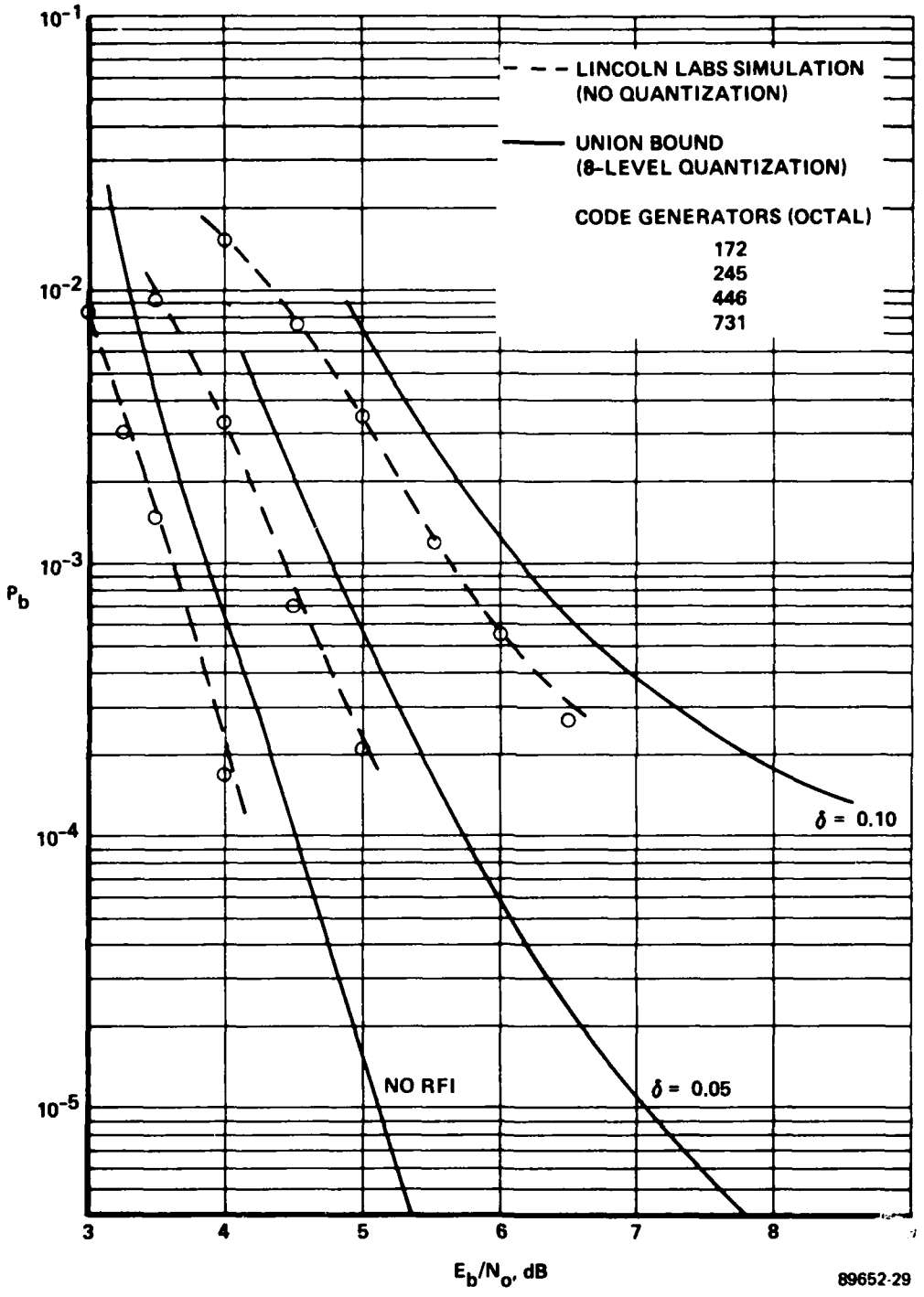


Figure 4.3.2-2. Performance of the R=3/4, k=9 Code Found by Paaske

8-level quantization, these curves will degrade by about 0.25 dB. We also show in this figure union bound calculations that we made assuming 8-level quantization and infinite path metric range. When the Lincoln Labs simulations are corrected by 0.25 dB they correspond very closely to our union bound estimates. Thus, these performance estimates assume no implementation degradation other than the use of 8-level quantization of the demod output.

We show similar performance curves in Figures 4.3.2-3 and 4.3.2-4 for the R=2/3(k=10) and R=3/4(k=11) codes. We have not completed an exhaustive search for the best codes at these two constraint lengths so we may get some additional improvement in performance. The R=2/3 and R=3/4 codes would probably only be necessary at the higher data rates that may eventually be used on shipboard. Therefore, they would most likely be used in severe RFI environment. Thus, maximization of coding gain in this environment is most important.

We show curves for δ=0.05 representing one interferer with 5% duty cycle and δ=0.10 representing two interferers each with 5% duty cycle. Note that with δ=0.10 performance is significantly degraded for the R=3/4 code. However, with δ=0.05 the code provides 3.5 dB of coding gain. With no RFI the gain on an AWGN channel is about 4.5 dB. For an environment with two interferers one could operate in the degraded mode, but where possible we would recommend using the R=2/3 code. This code provides 3.2 dB of coding gain in the presence of 10% RFI (δ=0.10). This code also provides 4.3 dB of coding gain with 5% RFI (δ=0.05) and 4.9 dB of coding gain on an AWGN channel with no RFI.

The performance curves we have presented have included all degradation in our decoder implementation. A contributor to error rate which has not been modelled entirely accurately is the "edge" effects of the WSC-3 blanker. Thus, a 5% duty cycle interferer does not cause exactly 5% erasures as we have modelled it. This will cause a small amount of additional degradation (a few tenths of a dB). In addition, as we discussed previously, the random erasure model that has been used does not exactly predict the performance when using a pseudorandom interleaver (the performance may be slightly

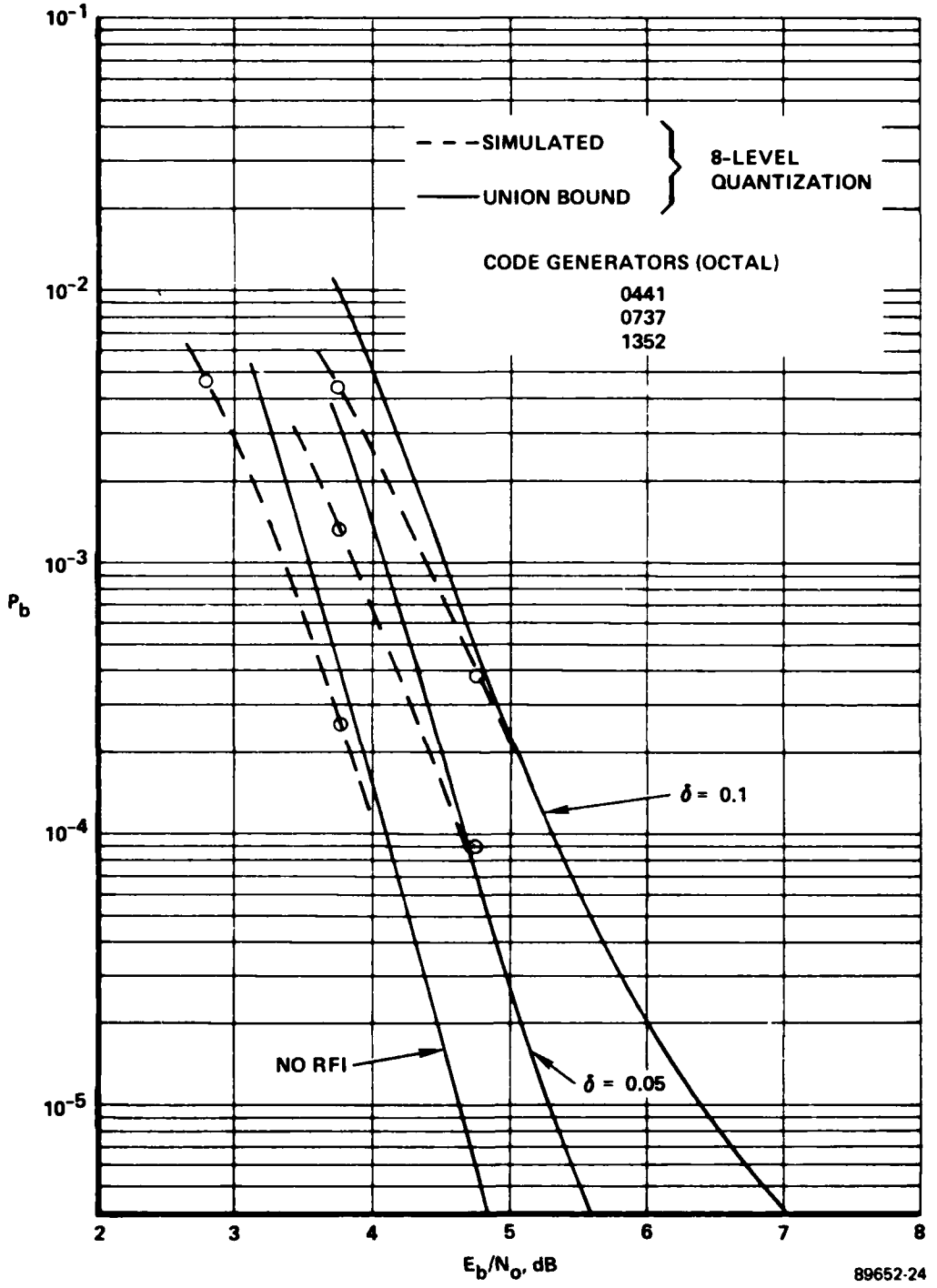


Figure 4.3.2-3. Performance of the R=2/3,k=10 Code

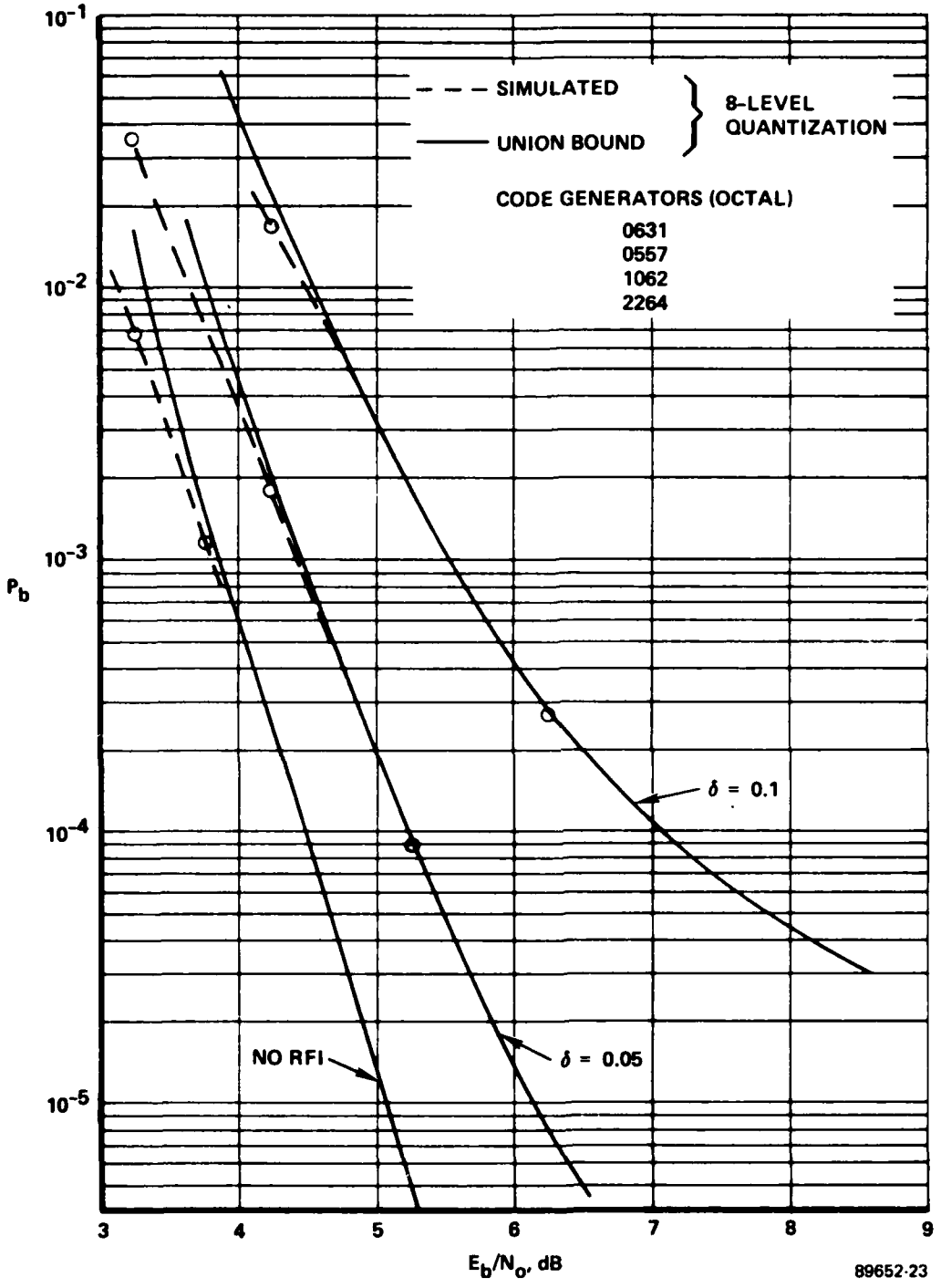


Figure 4.3.2-4. Performance of the R=3/4,k=11 Code

better or worse depending on the choice of interleaving sequence and RFI parameters). Finally, there are other sources of RFI which were not included in the model. All of these effects are rather tedious or impractical to evaluate through computer simulation, and for this reason we recommend that they be evaluated in Phase II using the RFI simulator.

The predicted coding gains at BER of 10^{-5} are summarized in Table 4.3.2-1. From these results we see that the gain produced by the R=1/2,k=9 code in the absence of RFI is sufficient to overcome the negative link margins noted in Section 2.1.1 for the FLTSATCOM SSIXS and TSCIXS links. As shipboard data rates are increased both RFI and insufficient link margin will become serious problems. Significant coding gain can be obtained at all code rates with 5% RFI. However, with 10% RFI gain can be obtained only with R=1/2 or 2/3. In the near term with the FLTSATCOM links being more power limited rather than bandwidth limited the R=1/2 code is recommended to allow the maximum increase in data rates (even with 10% RFI the coding gain is 4.5 dB). In the future as data rates are increased to the point where satellite bandwidth or modem transmission rates become limitations, then the R=2/3 and 3/4 codes will be more effective in achieving higher data rates. Note that the R=2/3 code will allow a 4.3 dB of coding gain with 5% RFI and 3.2 dB of gain with 10% RFI. This code will be very effective in improving link margins and thus allowing the use of higher data rates even in the presence of severe RFI. Finally, the R=3/4 code provides 3.5 dB of gain with 5% RFI so it will be very effective in improving link margins in the presence of moderate RFI environments. We also note that in comparing the gains of the longer constraint length codes we recommend with the gains of shorter constraint length Paaske codes in the presence of RFI that there is strong motivation for using the larger constraint length (a performance curve for the R=2/3,k=8 code has not been included but the coding gains are summarized in Table 4.3.2-1). The benefit of the larger constraint length is 1 to 2 dB depending on code rate and the amount of RFI. Since the decoder will be designed for relatively low data rates (32 kbps and lower) this increase in

Table 4.3.2-1. Predicted Coding Gains (in dB) at 10^{-5}

R	k	Code	No RFI	5% RFI	10% RFI
1/2	9	Larsen's	5.6	5.1	4.5
1/2	9	Larsen's	3.8*	3.4+	2.9+
2/3	8	Paaske's	4.7	3.6	1.3
2/3	10	Ours	4.9	4.3	3.2
3/4	9	Paaske's	4.5	2.6	---
3/4	11	Ours	4.5	3.5	---

*Hard decisions

+Hard decisions with blanking signal

constraint length can be accommodated without a large increase in decoder complexity.

4.4 References

4-1.

4-2. Richer, I., "Viterbi Decoding in the Presence of RFI-Part II: Interleaver," Lincoln Laboratory Memo, December 1976.

4-3. Knuth, D., The Art of Computer Programming, Vol. 2, pp. 9-19.

4-4. Larsen, K. J., "Short Convolutional Codes with Maximal Free Distance for Rates 1/2, 1/3, and 1/4," IEEE Transactions on Information Theory, Vol. IT-19, pp. 371-372, May 1973.

4-5. Paaske, E., "Short Binary Convolutional Codes with Maximal Free Distance for Rates 2/3 and 3/4," IEEE Transactions on Information Theory, Vol. IT-20, pp. 683-689, September 1974.

4-6. Richer, I., "Viterbi Decoding in the Presence of RFI-Part II: More Data, More Rates," Lincoln Laboratory memo, December 1976.

5.0 BRIEF DESCRIPTION OF PROTOTYPE DESIGN

In this section we present a brief description of the prototype design we are recommending for Phase II and a discussion of its relevance in various Navy SATCOM applications. This design is a very general-purpose test set for measuring the effectiveness of coded systems in a variety of potential Navy applications. We emphasize that only a fraction of all the functional pieces of this design will be required in any specific Navy application. We will demonstrate later exactly what is required for each of four important near-term potential applications.

5.1 Functional Description

A functional block diagram of our convolutional encoder/decoder test set interfaced with an AN/WSC-3 is shown in Figure 5.1-1. This test set can accept data at any data rate to 32 kbps, which is sufficient to cover all forseeable information for Navy use. On the transmit side, the encoder can encode incoming data with either a rate 1/2 (constraint length 9), rate 2/3 (constraint length 10), or rate 3/4 (constraint length 11) convolutional code. The encoded data is then pseudorandomly interleaved, and frame sync pulses are inserted. The resulting coded and randomized data can then be transmitted by the AN/WSC-3. On the receive side we are building an integrator and quantizer to take the I&D data from the AN/WSC-3 and provide 8-level soft decisions for the decoder. This allows an additional 2 dB of coding gain to be obtained. The de-interleaver contains a frame synchronizer which determines start of frame. Then the randomized data is reordered according to its original sequence and passed on to the decoder. The decoder performs maximum-likelihood decoding using the Viterbi algorithm for the selected code rate and passes the decoded data on to the user. The encoder/decoder may also be used without the interleaver/deinterleaver. In this case the decoder has a branch synchronizer which provides all necessary decoder synchronization (see Appendix B). When the encoder/decoder is used with the interleaver/deinterleaver the frame sync provides the necessary decoder sync and the branch

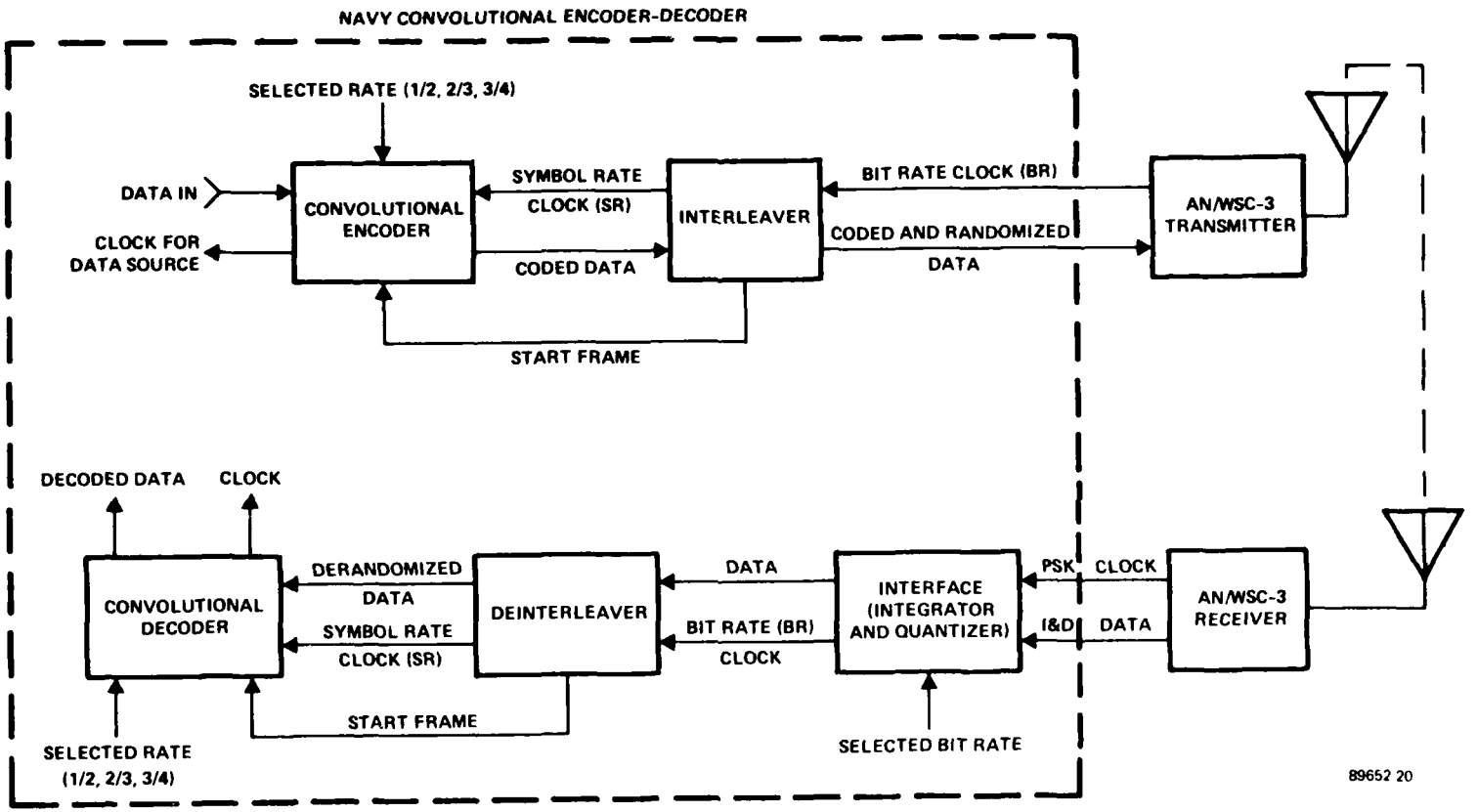


Figure 5.1-1. Test Set Functional Block Diagram

synchronizer is disabled. In addition, the integrator and quantizer may be bypassed either for operation with hard decisions or for operation with other modems that provide soft decision outputs. Though not shown in this block diagram a differential encoder/decoder is also included and may be used as desired. Differential encoding/decoding in the AN/WSC-3 is not employed.

5.1.1 Coder Implementation

We have chosen in our design to have three selectable code rates: 1/2, 2/3, and 3/4. In the 1/2 rate, for each data bit there are two output bits; in the 2/3 rate, for every two data bits there are three output bits; and in the 3/4 rate, for every three data bits there are four output bits. This allows us to accommodate a variety of signal-noise environments and bandspreading requirements. The encoder is simply a shift register into which data bits are shifted. Output bits are determined as linear combinations of preceding data bits. Generally, mod-2 adders are used to form the linear combinations of data bits necessary to generate the coded output bits. However, we have chosen a PROM rather than combinational logic to do this in order to more easily accommodate the multiple code rates we are providing. Differential encoding can optionally be switched in prior to the convolutional encoder, if desired.

5.1.2 Interleaver Implementation

The purpose of the interleaver is to randomize data according to a fixed algorithm so that if data is disturbed by a burst of noise when picked up by the receiver, the lost data is not really in one continuous chain. After de-randomizing in the deinterleaver by the same algorithm, the lost data will be randomly spread over the frame. This way, if only a few contiguous bits are lost in the received data, convolutional decoding can recover with high reliability the bits of data that were lost.

The deinterleaver needs to know where a frame starts in order to apply the de-randomizing algorithm in the right sequence to recover the original data. The decoder needs to know where each

coded group of bits starts as well. For these reasons a sync word made up of 36 bits is incorporated into two data frames, with the sync bits separated by 55 data bits. This way the deinterleaver can locate the beginning and end of the sync word (which is always a fixed format) and therefore the beginning end of each frame.

Data that comes to the interleaver is stored randomly in one RAM according to an algorithm stored in a PROM which is sequentially addressed by a state counter that counts up to the length of a frame. We have used a PROM to generate the randomizing sequence rather than a PN sequence to allow complete flexibility in the choice of sequence. During the next frame data is stored in the other RAM in the same manner as it was stored in the first RAM, while data that was stored in the first RAM is read out sequentially using the sequential addresses from the state counter.

One clock pulse every 56 pulses is removed from the bit rate clock (BR) before it is sent to the coder and data source. Therefore, the data that comes into the interleaver has a blank spot or hole every 56 bits where a sync bit (as part of the frame sync word) can be incorporated. The sync bits are stored in the PROM and are multiplexed into the stream of data every 56 bit rate clock pulses by a modulo 56 counter.

If the use of the interleaver is not desired, it can be bypassed by a switch in the front panel that will directly route the encoded data to the transmitter. This switch also prevents the sync clock pulse from being removed from the bit rate clock.

5.1.3 AN/WSC-3 Interface

Since the AN/WSC-3 modem provides only hard decisions, we propose to access an internal point in the modem and provide an external integrate-and-dump (I&D) circuit and 8-level quantizer. These soft decisions will allow us to achieve nearly an additional 2 dB of coding gain. We recognize that interfacing at this point may be suitable only for test purposes, and that for the present time an operationally satisfactory interface may be available only at the hard-decision AN/WSC-3 data output. However, we feel that the potential gain from soft decisions should be evaluated, especially

in view of the fact that the General-Purpose Modem currently under development will provide soft decisions.

Our implementation requires only two lines from the AN/WSC-3: the I&D data* from A1A7 P1-19 and the PSK clock from A1A7 P1-17. The I&D data from the AN/WSC-3 is routed to our integrator. After the I&D data is integrated it is routed to a quantizer where the analog signal is digitized. Thus the output of the quantizer is 2 lines representing soft decisions and a sign bit representing a hard decision. To insure that the proper relationship between the quantization levels and the integrator output is maintained, an AGC circuit is used to control the integrator gain. Our reasons for providing an external I&D rather than using the I&D circuit in the AN/WSC-3 are twofold. First, while the I&D input data is available externally from module A1A7, to access the I&D output data we would have to make a connection interval to module A1A3. Secondly, the AGC within the AN/WSC-3, while satisfactory for hard decisions, does not provide the proper scaling of I&D outputs into the quantizer for producing soft decisions; by controlling the integrator gain as we do, we insure the proper levels into the quantizer. This method of implementing the integrator and quantizer with the AGC loop has been used on various programs at Harris. Two of these are the MD-921 BPSK Modem which operates at any data rate from 1 kb/s to 10 Mb/s, and the MD-1002 BPSK/QPSK Modem which operates at any data rate from 16 kb/s to 20 Mb/s.

Since the clock supplied from the AN/WSC-3 is derived from the data, all that is necessary is to provide a means of properly aligning this clock and the dump circuit of the new integrator. This is a delay line and an inverter which we call Clock Control.

It should be noted that the I&D data received from the PSK detector A1A7 is <u>not</u> DC coupled. The significance of this fact is that biased data patterns - more ones than zeros or vice versa - will produce a DC offset resulting in a degradation to bit error rate performance <u>with or without</u> coding.

*Although identified as I&D data in the AN/WSC-3 documentation, this voltage is actually the <u>input</u> to the I&D circuit.

The interface implementation outlined above results in no modification to the AN/WSC-3 unit. The signal required from A1A7 can be obtained by tapping, in parallel, the original lines. In this way the original operation of the AN/WSC-3 will not be affected.

5.1.4 Deinterleaver Implementation

Randomized data that comes into the deinterleaver sequentially is sorted in such a way as to output sequential data in which all the bits are rearranged in the same order as they were before the interleaver randomized them. This function of the deinterleaver is very straightforward. The other function is to provide frame synchronization and this function requires a large part of the complexity of the deinterleaver. We are using a sync word distributed over two frames to indicate "start of frame". The frame synchronizer has two modes of operation. In the SEARCH mode the synchronizer has no information regarding the "start of frame" and must try all possibilities to locate the sync word. In the LOCK mode the synchronizer has previously located "start of frame" and simply continues to verify every two frames that the sync word is still located where it should be.

During the SEARCH mode each time a data bit is received it is assumed to be the end of a frame, and with this assumption the sync word is withdrawn from memory and checked. Each sync bit is picked from the memory by selecting its randomized address from the PROM. Then it is compared with what it should be (stored in the PROM as well) and for each mismatch an error is stored. The number of errors is compared to an established threshold, and if greater, then the mode stays in SEARCH. Otherwise it switches to LOCK.

During the LOCK mode data bits are written sequentially and read randomly from the memory at bit rate (BR). When sync bits are read, they are still compared with what they should be and every mismatch is stored as an error. If the errors stored for two frames exceed a threshold twice in a row, then the mode switches back to SEARCH.

The write in memory operation is always done in a ping-pong fashion; one frame is first written sequentially in one memory and then the next frame in the other memory, and so forth. The read operation is performed in two different ways. During the SEARCH mode frame bits are read randomly from both memories. This is because the search for the sync word starts at a different location in memory every time a new bit of data is written in, and the sync word is spread over two frames. The data bits have no significant until frame sync is achieved. During the LOCK mode data bits are read in a ping-pong fashion. While the newest frame is being written sequentially in one memory, the oldest one is being read randomly from the other memory.

The effect of the deinterleaver can be bypassed by a switch on the front panel by routing data and clock (BR) past the deinterleaver directly to the outputs.

5.1.5 Decoder Implementation

The Viterbi decoder portion of our hardware design is a modified version of a 100 kb/s, R=1/2,k=7 decoder initially developed by Harris ESD under IR&D funding, and now being built for NRL as part of an all-digital, low-cost modem/codec. The major modifications to this existing design are as follows:

- Variable code rates (1/2, 2/3, 3/4)
- Larger constraing lengths (k=9 for R=1/2,k=10 for R=2/3,k=11 for R=3/4)
- Extended decoding depth

These modifications will lead to somewhat greater complexity, primarily because of an increase in various memory requirements. This is due to an increase in the number of states from 64 to 256 and an increase in the decoding depth from 32 to 96. However, the basic logic and operations of the decoder will follow this proven design. Thus, the techniques used for branch synchronization, path metric updating and storage, information sequence updating and storage, and the technique for producing the output decisions. Rather than go into more detail on decoder operation at this point we have

chosen to include Appendix B which discusses the basic principles of maximum likelihood decoder operation.

5.2 Relevance to Potential Applications

The primary purpose of the test set that we recommend is to measure the effectiveness of coded systems in a variety of potential Navy applications. In order to adapt to various configurations, we have taken care to insure that the encoder/decoder can be used with or without interleaving, with or without the soft decision AN/WSC-3 interface (integrator and quantizer), and with or without differential encoding/decoding. Specific current and near-term applications which we can address include:

- Links to and from aircraft and submarines at low data rates (2400 bps), using an AN/WSC-3 or equivalent modem (SSIXS and TSCIXS).
- Low-rate link (2400 bps) to and from surface ships using an AN/WSC-3 or equivalent modem (CUDIXS, etc. al.).
- High-burst-rate TDMA links to and from surface ships.
- High-rate links to and from surface ships using a soft-decision modem such as the General-Purpose Modem under development.

Table 5.2-1 indicates part count for each of the functional elements in our proposed prototype unit, and shows which of these elements are required in each of the above applications.

The application of coding on the aircraft and submarine links is one of the most critical applications. As noted in Paragraph 2.1.1, these links have projected negative margin at 2400 b/s. Also, since there will be very little if any effect from RFI on these links, the largest coding gains can be obtained. Here we would recommend the R=1/2,k=9 code with no interleaving to give 5.6 dB of gain. These links can then be closed. Using our soft decision interface to the AN/WSC-3 results in a parts count of 133 IC's. If the integrator and quantizer were not used the parts count would drop to 113. This could be done if either the General-

Table 5.2-1.

Applications Matrix and Complexity Estimates

Function	Number of ICs	Application: 2400 bps A/C and SS Links	Application: 2400 bps Links to Ships	Application: High-Rate Shipboard TDMA Links	Application: High-Rate Links Using GP Modem
Integrator and Quantizer	20	X			
Interleaver/ Deinterleaver	50			X	X
Frame Sync	60				X
Branch Sync	13	X	X	X	
Encoder/ Decoder	100	X	X	X	X
TOTAL	243	133	113	150	210

Purpose Modem was used to interface to the AN/WSC-3 at IF* or if one simply used hard decisions from the AN/WSC-3 modem. Note that if hard decisions are used, 2 dB of the available coding gain is lost so we do not recommend this approach unless it can be demonstrated that the resulting gain is sufficient. We would also point out that this parts count is larger than would actually be required if only the rate 1/2 code were used as would be reasonable in this application. More path storage is required for rate 3/4 so 1/3 of the path memory IC's can be eliminated. Another alternative which could be used in this application is to shorten the constraint length to k=7, using the original design of the NRL encoder/decoder noted in Paragraph 5.1.3. This unit consists of two cards (approximately 80 IC's) in the hardware being delivered to NRL. This shorter code gives 5.0 dB of gain if soft decisions are used.

A similar equipment configuration is appropriate in 2400 bps links to surface ships, such as secure voice, CUDIXS, TADIXS and TACINTEL. In these links the margin problems are not as severe as in the aircraft and submarine links, but some RFI protection is desirable. Since RFI pulses are shorter than a bit time, interleaving is not required. Furthermore, since large coding gains are not required, it is reasonable to use a hard-decision interface to the AN/WSC-3, so that the integrator and quantizer can be omitted.

For high data rate shipboard applications the principal problem becomes providing RFI protection as well as some gain against thermal noise to maximize the achievable data rate. For both TDMA and non-TDMA applications modems are currently being built to interface at the AN/WSC-3 IF and provide soft decisions for the decoder. Thus, in these cases our integrator and quantizer will not be required. The test set has the capability of operation up to 32 kbps. The maximum envisioned data rate with coding is 24 kbps. This would be a 24 kbps information rate and a 32 kbps transmitted bit rate (with rate -3/4 coding).

*We note that in the TSCIXS application, the intention is to employ the soft-decision GP Modem interfaced to an AN/ARC-143B radio.

In a TDMA application no frame synchronizer would be required since TDMA frame sync is available. This is a significant parts savings in the interleaver/deinterleaver. In addition, no decoder branch synchronization is required since this information is also available from the TDMA frame sync. The parts count for this application is 150 IC's. The codes available are rates 1/2, 2/3, and 3/4, offering successively smaller values of gain and reduced bandspreading.

In non-TDMA high data rate applications we have the highest parts count - 210 IC's. This is due to the inclusion of the frame synchronizer. We chose to include a frame synchronizer in the test set to provide rapid frame acquisition time (about 1000 bits). However, we could have eliminated the frame synchronizer and relied on the decoder branch synchronizer to do the necessary searching and detection of proper deinterleaver sync. This would result in much greater acquisition times (about 500,000 bits). By using this approach one could effect considerable savings in complexity at the expense of a long acquisition time, but we do not regard this as desirable in tactical applications.

In summary, we note from Table 5.2-1 that although the proposed test set has a rather large parts count (243 IC's), the configuration of these functional elements which is appropriate for any specific application, even using the same means of implementation, would have significantly reduced parts count. Moreover, when designing operational hardware for a specific application, additional economies are available by reducing the flexibility we have designed into each functional element (e.g. multiple-rate codes) and by the use of custom LSI techniques. Thus any final design will be considerably smaller than the general-purpose test set, and well in line with the size, weight, and power constraints imposed by the platform characteristics.

6.0 CONCLUSIONS AND RECOMMENDATIONS

In the foregoing sections of this report we have established the following major points:

- Coding can provide significant benefits in the Navy UHF satellite communications environment.
- Among state-of-the-art coding techniques, the use of interleaved convolutional codes with Viterbi decoding is superior to other coding techniques applicable to this problem.
- This recommended technique can satisfy both the technical and operational requirements of Navy SATCOM systems.
- Certain applications will require less than the complete suite of functional elements recommended in Section 4 and accordingly will be simpler in hardware.

Reasonable extensions of these efforts, in our judgment, should be in two major areas. The first of these is verification that the predicted performance can actually be approached in a realistic environment. Our predictions generally ignore modem implementation losses, non-ideal blanker performance, etc. Along the same lines, we have not quantitatively evaluated performance against non-radar forms of RFI such as harmonics and intermodulation products, although we contend that coding gain helps overcome the effect of receiver desensitization. This first area of investigation will be accomplished by the fabrication of a prototype or test set and by a comprehensive test and evaluation program using the RFI simulator at the Naval Postgraduate School.

The second area of extension of this work which we recommend is to combine the results of the evaluation program at Monterey with the most up-to-date projections of the technical and operational characteristics of Navy SATCOM Links (satellite characte tics, platform antenna gains, required data rates, etc.), to refine the recommendations as to what kind of coding equipment should be procured for various Navy applications. The end product of this reevaluation would lead to a collection of several functional specifications on coding equipment, one for each of several basically different applications (e.g., 2400 bps submarine communications, high

rate shipboard TDMA links, etc.) within the general Navy UHF SATCOM problem.

In our judgment, carrying out these two objectives in Phase II of the program is the most technically sound, cost-effective, and timely means to ultimately achieve the goal of procuring adequate coding equipment for use in the Fleet.

APPENDIX A

THEORY AND PERFORMANCE OF CONVOLUTIONAL CODES

A1.0 INTRODUCTION

The use of convolutional coding with maximum likelihood decoding has recently found wide application in communication systems. In most cases this has been shown to be the most practical coding technique for achieving large power gains (on the order of 5 dB). The purpose of this appendix is to present a tutorial discussion of the theory of convolutional codes. Topics to be discussed include convolutional code structure and performance.

A2.0 CONVOLUTIONAL CODE STRUCTURE

A constraint length k convolutional encoder consists of a k-stage shift register with selected stages being added modulo 2 to form the encoded bits. Consider the simple rate $\frac{1}{2}$* convolutional coder shown in Figure A-1. A convenient way to consider the relationship between the input and output sequences of such a coder is to utilize the "tree" shown in Figure A-2. The convention used is that an input 0 causes us to take the upper branch and an input 1 causes us to take the lower branch. Thus, any input sequence traces out a particular path through the tree. Specifically, a 10110 input sequence causes a 1101001010 output sequence. Of course, as the input sequence grows in length, the number of possible paths grow exponentially limiting the role of such a tree diagram to a conceptual one.

An optimum decoder, after reception of the signal, will choose the possible output sequence which correlates best with the received waveform and choose the input sequence which corresponds to that path through the tree. One might deduce that selection of a sequence that departs from the correct sequence at a particular node in the tree would result in a sequence incorrect from that point on producing an arbitrarily large number of bit errors. Fortunately, error events have a very short duration for properly chosen codes due to the code structure.

*Rate ½ implies that two output bits are generated in the coder for each input bit.

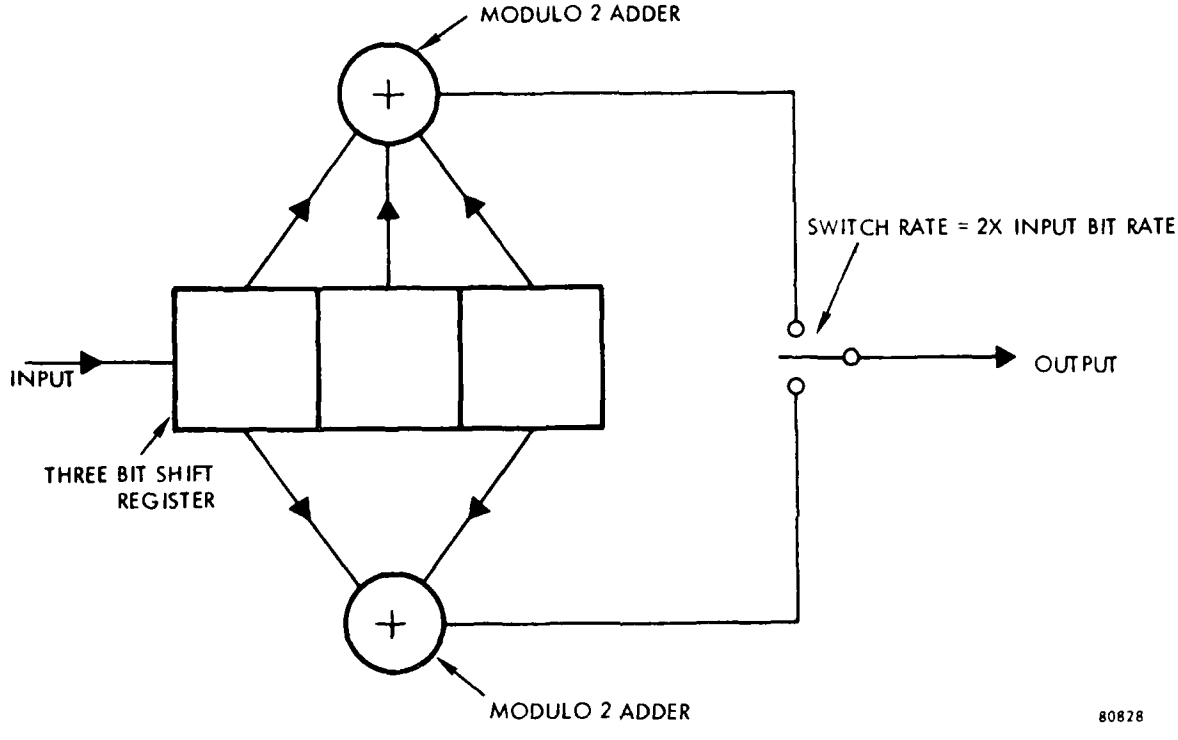


Figure A-1. Encoder for a Constraint Length 3 Convolutional Code

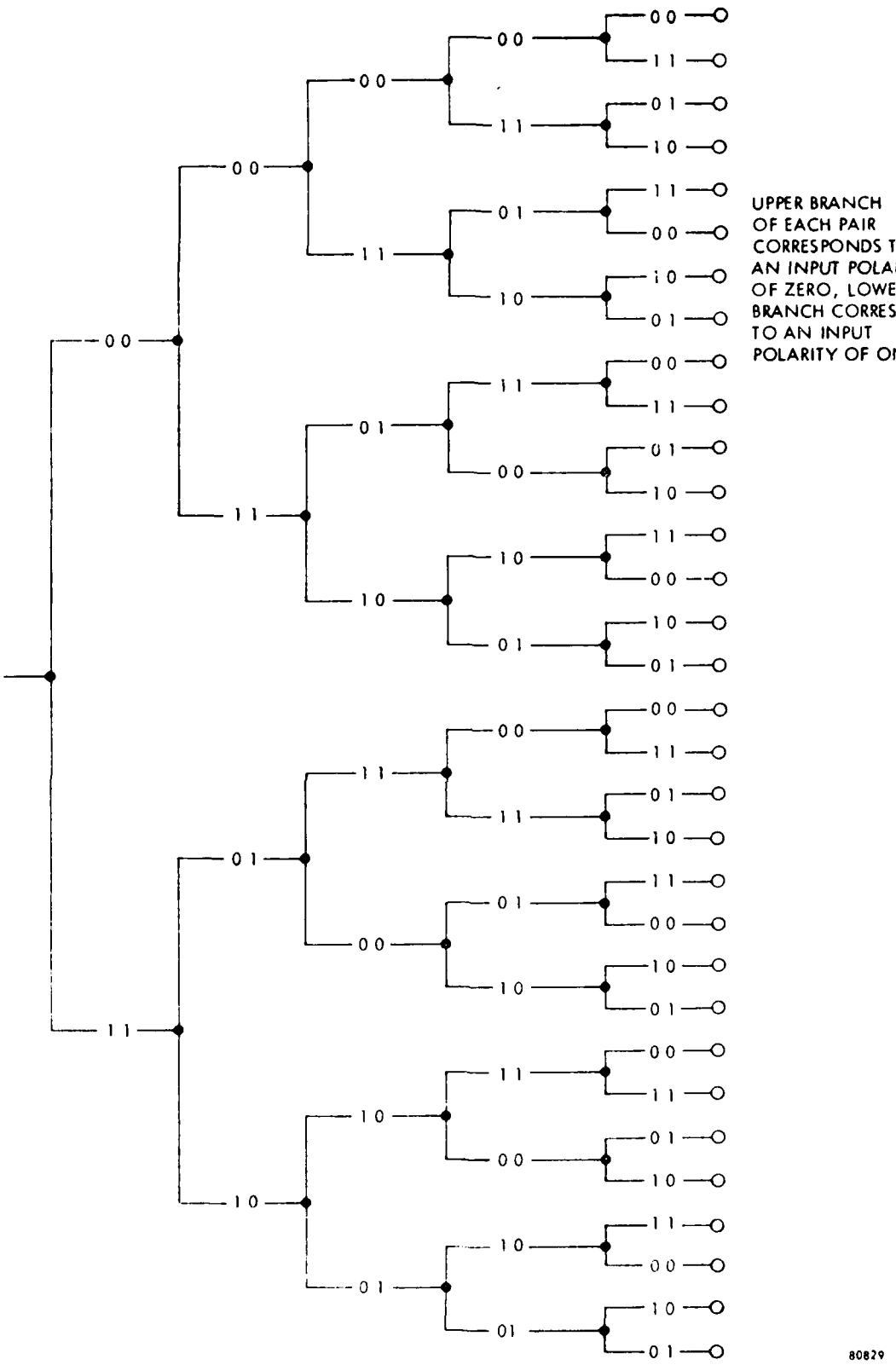


Figure A-2. Tree for Constraint Length 3 Convolutional Code

A2.1 Trellis Representation of Convolutional Codes

Figure A-3 represents an alternate way of viewing the same code tree of Figure A-2 and has been called a trellis structure by Forney[A-1]. As in Figure A-2, the convention is that an input 0 corresponds to selection of the lower branch. As before, each possible input sequence corresponds to a particular path through the trellis. For instance, an input sequence of 10110 can be seen to provide an output sequence of 1101001010 which is identical to the result that was obtained from Figure A-2. The significance of the trellis viewpoint is that the number of nodes in the trellis does not continue to graw as the number of input bits increases, but remains at 2^{k-1} where k is the constraint length of the code (the number of shift register stages necessary in the coder). Thus if an incorrect decision is made at a particular node the correct path may be regained later. That is, if the shift register hookups are chosen properly and noise causes an incorrect path to be more probable than the correct path, with high probability it will differ from the correct path over a short interval only and not for the remainder of the transmission.

These codes are easily generalized to rates other than ½. For rate 1/n convolutional encoders there are n modulo-2 adders and n output bits for each information bit fed into the encoder. Rate m/n codes can be implemented in exactly the same fashion except that input bits are fed into the register in groups of m rather than one at a time so that n channel bits are generated for each group of m information bits.

A2.2 Finite State Machine Representation of Convolutional Encoders

For rate 1/n codes the state of the convolutional encoder is determined by the most recent (k-1) information bits shifted into the encoder shift register. The state is assigned a number from 0 to $2^{k-1}-1$ which is obtained from the binary representation of the most recent (k-1) information bits with the most recent bit being the least significant bit. Given the state that the encoder is in and the new input information bit, the state to which the encoder proceeds and the encoded channel bits that result are determined. A complete description of the encoder as far as input and resulting output are concerned can be obtained from a state diagram like the one shown in Figure A-4 for the k=3, rate 1/2 code of Figure A-1. The bit along the top of the transition line indicates the information bit input to the decoder

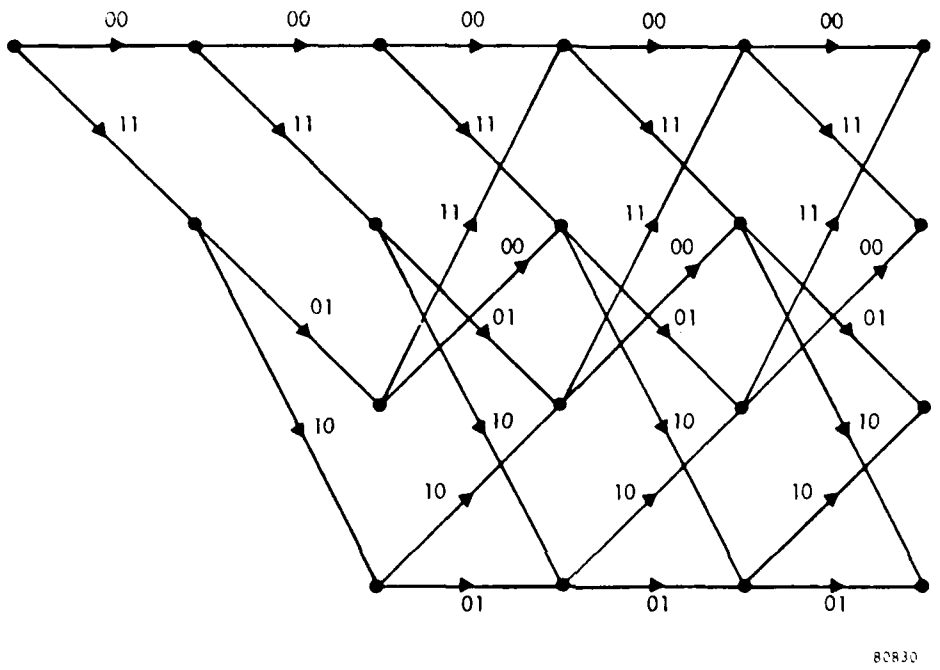


Figure A-3. Trellis for a Constraint Length 3 Convolutional Code

that causes the transition and the bits below the transition line show the resulting channel bits outputs from the encoder. Any sequence of information bits dictate a path through the state diagram and the channel bits encountered along this path constitute the resulting encoded channel sequence.

A similar description exists for rate m/n codes. In this case there are k-m states and 2^m transitions are possible from each state. The current state is determined by the most recent k-m bits and the transition out of this state is determined by the m new information bits. For rate 1/n codes and rate m/n codes the topological form of the state diagram depends only on the constraint length and m and not on the precise hookups that are used. In general the new state number is related to the old state number by

$$N_{new} = (2^m N_{old} + P) \text{Mod } 2^{k-m} , \qquad (A\text{-}1)$$

where P is an m bit binary number that is numerically equal to the m new information bits with the most recent bit being the least significant digit.

Due to the topological equivalence between the state diagram of Figure A-4 and a signal flow graph (Mason and Zimmerman [A-2]) one is tempted to ask the question of whether the properties and theory of signal flow graphs might profitably be applied to the study of convolutional codes. In fact, signal flow graphs have been found to be a very useful tool in analyzing convolutional code structure and performance [A-3, A-4]. Any quantity that can be accumulated linearly on a per branch basis can be computed using a signal flow graph by writing that quantity as the exponent of some indeterminate and letting the result be the gain of the branch. The gain of the overall graph between two states of interest will then provide a generating function for the quantity under consideration. (Note that a necessary condition for obtaining a generating function is that none of the loops possess unity gain. This condition is normally avoided for reasons which will be discussed.) Thus generating functions for such quantities as weight, length, transitions, etc., can be computed using the signal flow graph. Since we may consider the all zero code sequence as the transmitted one without loss of generality, we are especially interested in those paths which start and end in the zero state and do not return to the zero state anywhere in between. For each of these paths it is desirable to determine the length of the input sequence, the weight

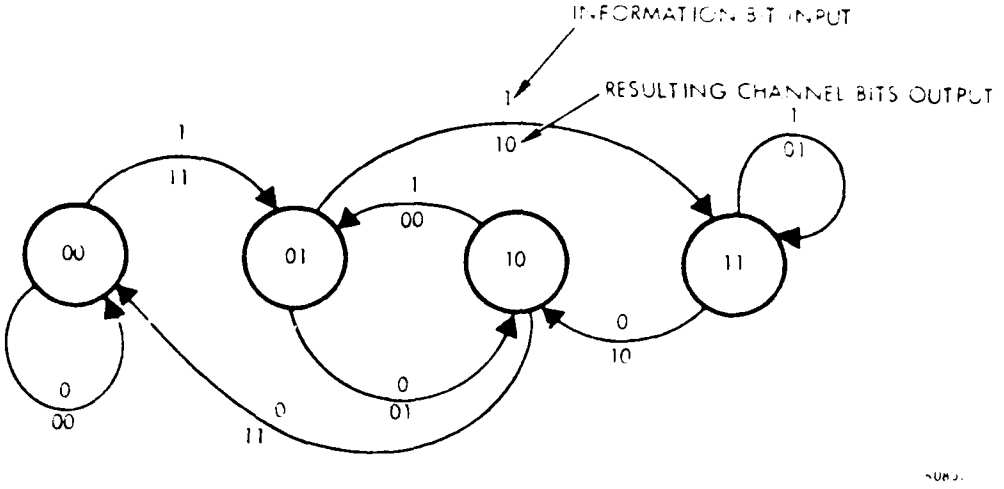


Figure A-4. State Diagram for a k=3, Rate 1/2 Convolutional Code

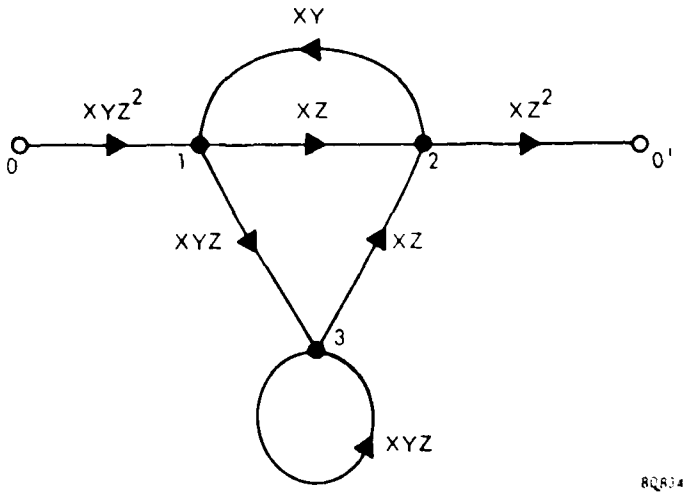
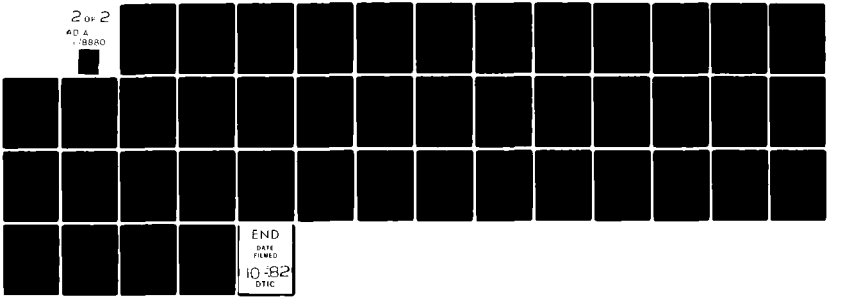


Figure A-5. Signal Flow Graph for the Convolutional Code of Figure A-4.

AD-A118 880 HARRIS CORP MELBOURNE FL GOVERNMENT ELECTRONIC SYSTE--ETC F/G 17/2
CODING FOR NAVY UHF SATELLITE COMMUNICATIONS (PHASE I).(U)
APR 77 N00039-76-C-0384
UNCLASSIFIED JA-6866 NL

2 OF 2
AD A
118880



END
DATE FILMED
10-82
DTIC

of the input sequence, and the weight of the output sequence. Such a determination can be made using signal flow graphs as illustrated in the following example.

Let x be the indeterminate associated with the length of the input sequence, y be the indeterminate associated with the weight of input sequence, and z be the indeterminate associated with the weight of the output sequence. Each branch in the flow graph will have a gain g given by

$$g = xy^{\omega_i} z^{\omega_o} , \tag{A-2}$$

where ω_i is the weight of the input required to drive the coder between the two nodes connected by the branch and ω_o is the weight of the output associated with the branch. Now consider the convolution code of Figure A-1 whose state diagram is shown in Figure A-4. Since we are only interested in those paths that start and end in state zero and do not go through zero in between, we remove the self loop on the zero state and split the zero state into an input and an output. The resulting flow graph for this code is shown in Figure A-5. When the gain of this graph is computed using standard methods (Mason and Zimmerman[A-2]),

$$G = \frac{x^3yz^5}{1-x^2yz-xyz} . \tag{A-3}$$

When expanded by long division

$$\begin{aligned} G = {} & x^3yz^5 + (x^4+x^5)\, y^2z^6 \\ & + (x^5+2x^6+x^7)\, y^3z^7 \\ & + (x^6+3x^7+3x^8+x^9)\, y^4z^8 \\ & + \text{etc.} \end{aligned} \tag{A-4}$$

Thus there is one path of input weight 1, output weight 5, and length 3. There are two paths of input weight 2, output 6, and lengths 4 and 5. There are four paths of input weight 3 and output weight 7. One path has length 5, two have

length 6 and one has length 7. This information determines the structure of the code which in turn determines the code performance.

A2.3 Code Generator Polynomials

The outputs of a convolutional encoder can be described as a polynomial multiplication of the information input and the generators of the code. For example, the k=3, rate 1/2 encoder of Figure A-1 has two outputs and two generators. The upper generator polynomial is $1+x+x^2=g_1(x)$ and the lower generator polynomial is $1+x^2=g_2(x)$. We can express the information bit sequence in terms of a polynomial $I(x)=i_0+i_1x+i_2x^2+i_3x^3+\ldots$, where i_j is the information bit (0 or 1) at the jth bit time. The output of the upper encoded channel sequence of Figure A-4 can then be expressed as $T_1(x)=I(x)g_1(x)$ and the lower by $T_2(x)=I(x)g_2(x)$ where the polynomial multiplication is carried out in GF(2).

For rate m/n codes this description is similar only slightly more complicated. In this case the input is subdivided into m subsequences which are represented by polynomial $I_1(x),I_2(x),\text{---},I_m(x)$. The output sequences are then determined by equation of the form

$$T_i(x) = \sum_{p=1}^{m} g_{ip}(x)I_p(x) \ . \qquad \text{(A-5)}$$

Consequently in the general case the input and output sequences are related by an n and by m generator matrix that contains nm subpolynomials as elements.

The code generator polynomials are usually chosen so as to minimize the average probability of error on the output of the decoder for a particular signal-to-noise ratio and for a specified decoding span. There are, however, two other code characteristics which enter into the code selection. One characteristic is that improperly chosen code generators can cause catastrophic error propagation. The other characteristic is that for most code rates and constraint lengths it is possible to choose the generator such that the code is transparent to phase inversions. Both of these effects occur as a result of the linearity of the convolutional codes. That is if $I_a(x)$ produces a code sequence $C_a(x)$ and if $I_b(x)$ produces a code sequence $C_b(x)$ then $I_a(x)+I_b(x)$ will produce a code sequence $C_a(x)+C_b(x)$.

In cases where catastrophic error propagation occurs there exists certain semiinfinite input sequences containing a large number of "ones" that produce a code sequence which contain a small number of "ones". Thus a small number of errors occurring in the communication channel can transform a particular code word into a second code word for which the corresponding input sequence differs in a very large number of positions. This effect is discussed in considerable detail in both the Interim Report on Convolutional Coding Research[A-3] and by Massey and Sain[A-5] who first provided a formal explanation. For rate 1/n codes the cure is to avoid generators which have a common factor. For rate m/n codes one avoids generators such that the $\binom{n}{m}$ determinants formed from the $\binom{n}{m}$ distinct submatrices of the n by m generator matrix do not have a common factor.

In order for a code to be transparent to a phase inversion it is necessary for the all "ones" input sequence to produce the all "one's" code sequence. Since inverting the channel sequence is identical to adding an all "one's" sequence, the linearity of the code guarantees that this will also invert the input sequence. For rate 1/n codes a necessary and sufficient condition for the code to be transparent to phase inversions is for the generator polynomials to each be of odd weight. One can easily verify by inspection that this will produce the necessary effect. For rate m/n codes the requirement is similar but slightly more complex.

A3.0 PERFORMANCE OF CONVOLUTIONAL CODES

The most useful techniques for estimating the performance of convolutional coding are union bounds and computer simulation. The usefulness of computer simulation is limited by the long computation times that are required to get a good statistical sample (it may take several hours to get a single point). Consequently, the greatest usefulness of this technique is in special coding applications where union bounding cannot be used to provide a good performance estimate. On the other hand, union bound estimates may be obtained through the use of a small amount of computer time. The union bound approach provides an estimate of probability of bit error (assuming an optimum decoder) which is accurate to within a small fraction of a dB for all signal-to-noise ratios large enough to give an error rate of 10^{-3} or less. In addition, this approach may also be used to predict performance for some specific hardware

configurations (used in implementing the decoder). The use of this technique is discussed elsewhere in more detail [A-3, A-4]. However, we shall provide a brief discussion of this technique below.

Since the codes considered are linear codes, the all zeros path may be assumed to be the transmitted path without loss of generality. A first-event error is made at the i'th received branch if the all zeros path is eliminated at this point by another path merging with it. The probability that the all zeros path is eliminated by a path merging with it is dependent only on the weight of that path. Denote this probability for a weight j path by P_j. The number of possible paths of weight j merging with the all zeros path is denoted by n(j), and the total information weight of these paths is denoted by w(j). The functions of n(j) and w(j) describe the structure of the particular code. The first few terms of the structure for the R=3 example code were found in Section A2.2. The weight structure for a large number of short constraint length codes is given in [A-3].

A union bound on the probability of a first-event error P_E, at branch i may be obtained by summing the error probabilities for all possible paths which merge with the all zeros path at this point. This overbound is then simply

$$P_E < \sum_{j=0}^{\infty} n(j)P_j \ . \qquad \text{(A-6)}$$

A union bound on the probability of bit error, P_B, may be obtained from (A-6) by weighting each term by the corresponding number of bit errors (the information weight for each path). However, for a R=m/n code there are m bits decoded on each branch. Thus, P_B is bounded by

$$P_B < \frac{1}{m} \sum_{j=0}^{\infty} w(j)P_j \ . \qquad \text{(A-7)}$$

The values for P are dependent on the type of channel. For the BSC and odd values of path weight, j, P_j is simply the probability that there will be channel errors in more than half the nonzero positions of the codeword being considered. For even values of j, P_j has an additional term equal to one-half the probability that there will be channel errors in exactly one-half

of the nonzero positions (since in this case the choice between the all zeros word and the competing word must be made randomly). Thus,

$$P_j = \begin{cases} \sum_{i=\frac{j+1}{2}}^{j} \binom{j}{i} p^i (1-p)^{j-i}, & j \text{ odd} \\ \frac{1}{2}\binom{j}{j/2} p^{j/2}(1-p)^{j/2} + \sum_{i=j/2+1}^{j} \binom{j}{i} p^i (1-p)^{j-i}, & j \text{ even.} \end{cases} \quad \text{(A-8)}$$

For PSK signaling on an additive white Gaussian noise (AWGN) channel, P_j is given by

$$P_j = Q(\sqrt{2jRE_b/N_o}) \ , \quad \text{(A-9)}$$

where

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \, dx \ . \quad \text{(A-10)}$$

In this equation E_b is the energy per information bit and N_o is the noise spectral density. This calculation assumes no quantization of the demodulator output, and this is the calculation that is usually used in estimating code performance. Typically in using 8-level quantization of the demod output the performance is about 0.25 dB worse than that given by (A-7) and (A-9), and the resulting performance estimates are corrected by this amount.

However, rather than take this approach in computing a union bound for 8-level quantization we have taken a direct approach. We have directly calculated P_j for 8-level quantization for use in the union bound (A-7). This is done as follows.

The quantization of the demod output into 8 levels numbered 0,1,...,7 is done as shown in Figure A-6. The bit metric assignment assumed for a received quantization level q is

$$m_o = 7-q \quad \text{(A-11)}$$

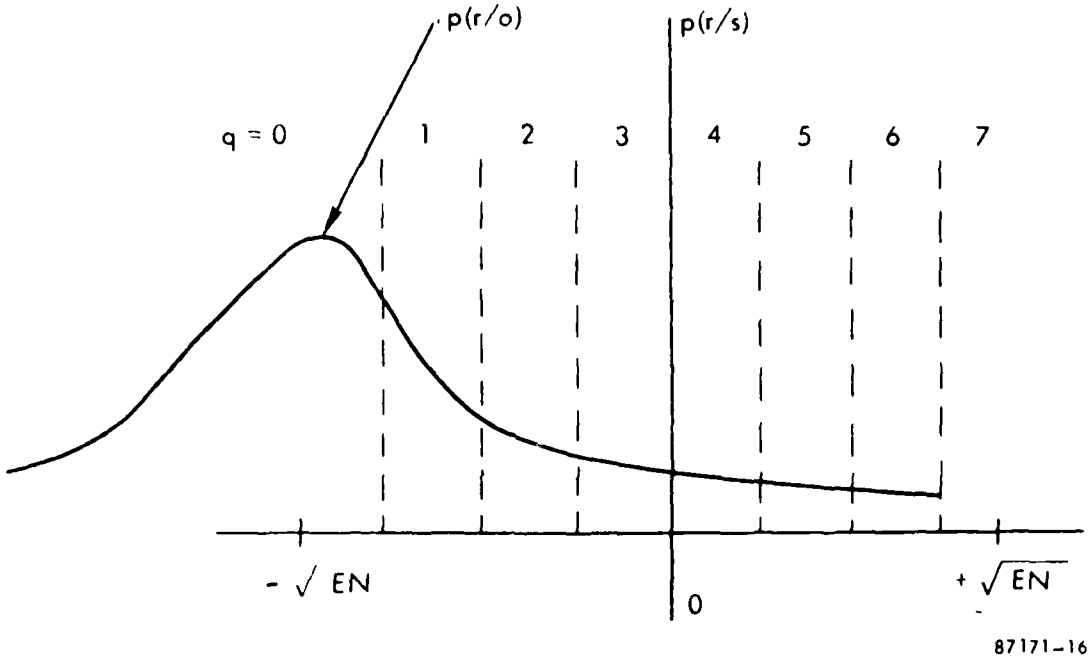


Figure A-6. Quantization Levels for the Demodulator Output Voltage

for every zero and

$$m_1 = q \tag{A-12}$$

for every one on the path being considered. The bit metric increments for an all zero path and a nonzero path differ only in those positions of the nonzero path corresponding to "ones". These are the only positions which cause the path metrics of these two paths to differ. The difference in the metric increments at these positions is

$$\begin{aligned} m_d &= m_o - m_1 \\ &= 7 - 2q \ . \end{aligned} \tag{A-13}$$

The set of m_d's is simply $\{-7,-5,-3,-1,+3,+5,+7\}$. This set has a probability density function, $p(m_d)$, which is identical to the probability of occurrence of the 8-levels, q, for a transmitted zero since the all zero transmitted sequence is assumed. Thus, the difference in metric increments for each position where two codewords differ is a random variable with probability density function $p(m_d)$. If the two sequences differ in j positions with a memoryless channel, the total metric difference is the sum of j identically distributed, independent random variables. Denote the total difference by t_d. The probability density function of t_d can be obtained by convolving $p(m_d)$ with itself j-1 times, i.e.,

$$p_j(t_d) = \underbrace{p(m_d)*p(m_d)*\ldots*p(m_d)}_{j \text{ terms}} \ . \tag{A-14}$$

The probability that the incorrect path is selected is simply the probability that t_d 0 plus one-half the probability that $t_d = 0$, i.e.,

$$P_j = \tfrac{1}{2}\, p_j(0) + \sum_{i=-\infty}^{-1} p_j(i) \ . \tag{A-15}$$

The discussions assumed certain values of assigned metrics as given in (A-11) and (A-12). However, the technique is not restricted to these assignments and can use any set of metric assignments. We have written a computer program for evaluating the union bound (A-7). The resulting performance curves for the best R=1/2 codes with 2-level (Q=2) and 8-level (Q=8) quantization for several constraint lengths up to k=7 are shown in Figure A-7. These curves assume binary, coherent PSK signalling with additive Gaussian noise. In addition, the performance curve is shown for a k=7 decoder using a proprietary implementation technique developed at Harris Corporation for handling the branch and path metrics which significantly reduces decoder complexity. We note that by using a k=7 code at 5.0 dB reduction in the E_b/N_o required to achieve $P_B=10^{-5}$ is obtained. It is rather surprising that such a large coding gain can be achieved with such a short constraint length. However, the gain is very real and can be used to decrease the required transmitted power by 5 dB for the desired information rate. The encoder configuration for this code is similar to that in Figure A-1 except that a seven stage register is required. Thus, the encoder complexity is typically very small. Most of the complexity in applying this technique resides in the maximum likelihood decoder.

REFERENCES

A-1. Forney, G. D., "Review of Random Tree Codes," Appendix A, Study of Coding System Design for Advanced Solar Missions, NASA Contract NAS2-3637, Codex Corporation, December 1967.

A-2. Mason, S. J. and Zimmerman, H. J., Electronic Circuits, Signals, and Systems, Wiley, 1960.

A-3. Clark, G. C., Davis, R. C., Herndon, J. C., and McRae, D. D., "Interim Report on Convolution Coding Research," Memorandum Report No. 38, Advanced Systems Operations, Radiation Systems Division, September 1969.

A-4. Viterbi, A. J., "Convolutional Codes and Their Performance in Communication Systems," IEEE Trans. on Communication Technology, Vol. COM-19, October 1971, pp. 751-772.

A-5. Massey, J. L. and Sain, M. K., "Inverses of Linear Sequential Circuits," IEEE Trans. on Computers, Vol. C-17, April 1968, pp. 330-337.

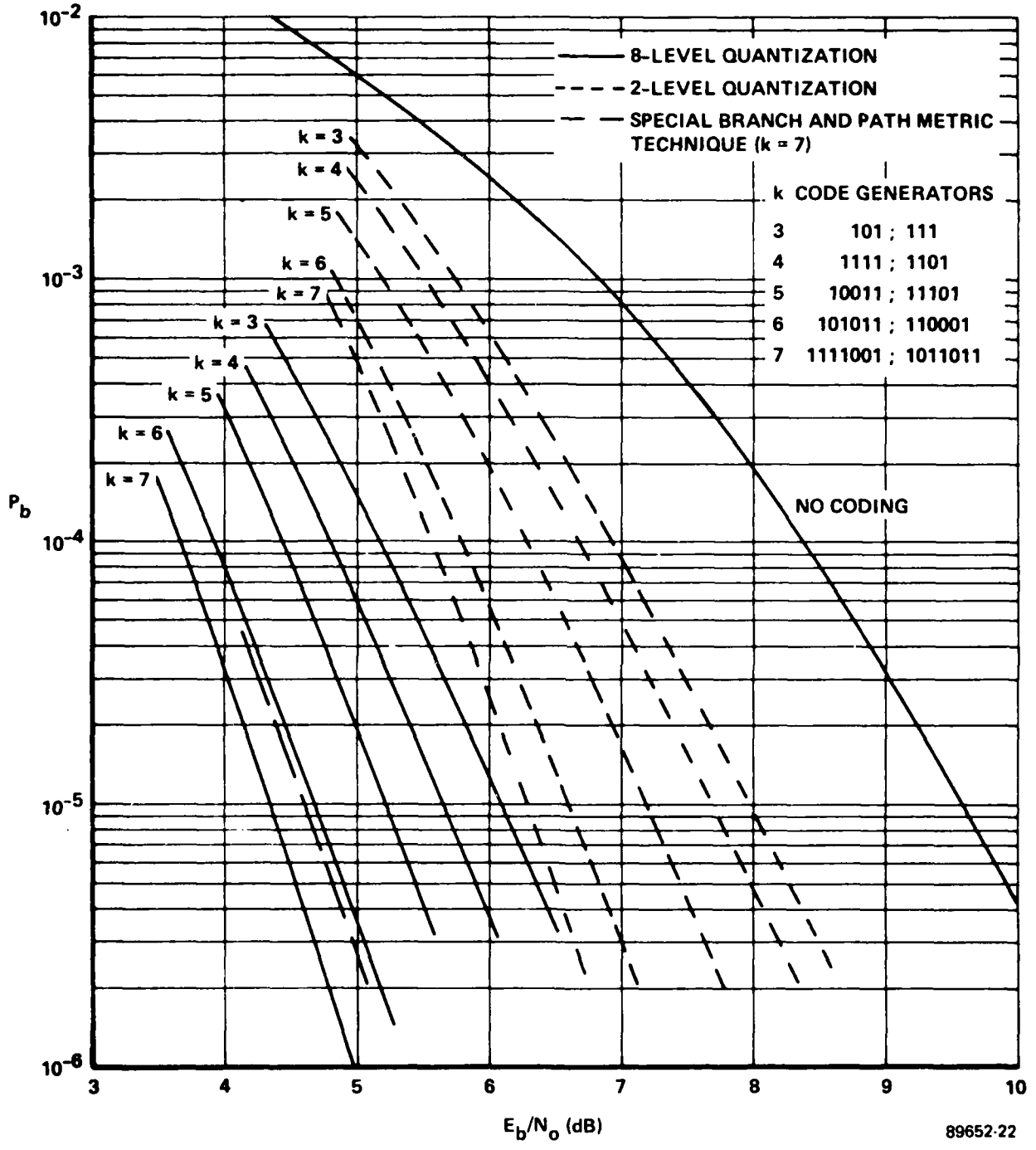


Figure A-7. Performance of Rate -1/2 Convolutional Codes with Maximum Likelihood Decoding

APPENDIX B

IMPLEMENTATION OF MAXIMUM LIKELIHOOD DECODERS FOR CONVOLUTIONAL CODES

B1.0 INTRODUCTION

The theory and performance of convolutional codes was discussed in Appendix A. The performance calculations that were presented were performed assuming that a maximum likelihood decoder was used. In this appendix the maximum likelihood decoding algorithm and decoder implementation considerations will be discussed.

B2.0 MAXIMUM LIKELIHOOD DECODING ALGORITHM

The maximum likelihood decoding algorithm for convolutional codes was discovered by Viterbi[B-1]. Maximum likelihood decoding implies that the decoder chooses the sequence out of all possible transmitted sequences which correlates best with the received sequence. However, the unique feature of the Viterbi algorithm is that the number of possible transmitted sequences that must be examined is limited by using the merging property of convolutional codes. That is, when two paths merge at a given node in the trellis, only the path with the larger correlation with the received sequence need be examined further. The reason for this is that the two paths will be identical from that point on since they are starting from the same state, and the path with the larger correlation at merging will always have the larger correlation. Thus, if every time two paths merge the path with the smaller correlation with the received sequence is discarded, then no path is discarded which has a chance of being the path with the largest correlation. Maximum likelihood decoding can then still be performed even though certain paths are discarded as described above.

A good conceptual picture of the operation of the Viterbi algorithm decoder can be obtained from the trellis diagram description of a convolutional code given in Figure A-3. Note in this figure that there are two paths leading into each of the 2^{k-1} nodes at a branch of the diagram. The decoder computes the correlations for each of these paths and discards that path having the smaller correlation at a node. This operation can be pictured as erasing the

corresponding node to node connection in the trellis diagram. At each time increment we can draw an incomplete trellis diagram such as the hypothetical case shown in Figure B-1. Only the connections corresponding to the larger correlated paths into each node are retained. One can think of the decoder as outputting 2^{k-1} of these trellis connections at each branch time telling how to draw the incomplete trellis diagram.

Note in Figure B-1 that the diagram graphically demonstrates how each path fares as additional branches are input to the decoder. Some paths stab off in various directions only to be eliminated as additional inputs are available to the decoder. Note also in the diagram that the 2^{k-1} survivor paths tend to eventually all lead back to a unique node representing the state in which the decoder estimates as most likely for the encoder at that unique node's branch. In this way the Viterbi algorithm inherently makes bit decisions, i.e., all survivor paths tend to lead back to a unique node (and hence to a unique information bit). This "merging" of the survivor paths cannot be guaranteed at a given decoding depth but if the decoding depth is made large enough the merging occurs with high probability. Various schemes for choosing the bit to be decoded from the survivor path are possible. One way is to output the oldest bit in the highest correlated path. Another technique is to output the bit that is in the majority at the oldest bit position of the 2^{k-1} survivor paths. Still another possibility is to output the oldest bit of an arbitrary, fixed survivor path in state 0. It can be shown that if the decoding depth is chosen large enough ($\geq$4 to 6 constraint lengths for R=1/2 codes) the error correction performances for all the techniques are identical.

B3.0 IMPLEMENTATION CONSIDERATIONS

A functional block diagram for a general Viterbi decoder is shown in Figure B-2. It accepts at the input a continuous sequence of m-bit binary numbers which represent the successive outputs of the channel bit detector quantized to one of 2^m levels and delivers the decoded information stream at the output. The most common soft decision approach is m=3 giving 8 levels. The machine itself consists of (1) a synchronizer which provides both branch timing information to the decoder and determines the correct bit polarity if this is required, (2) a branch metric computer which determines the appropriate

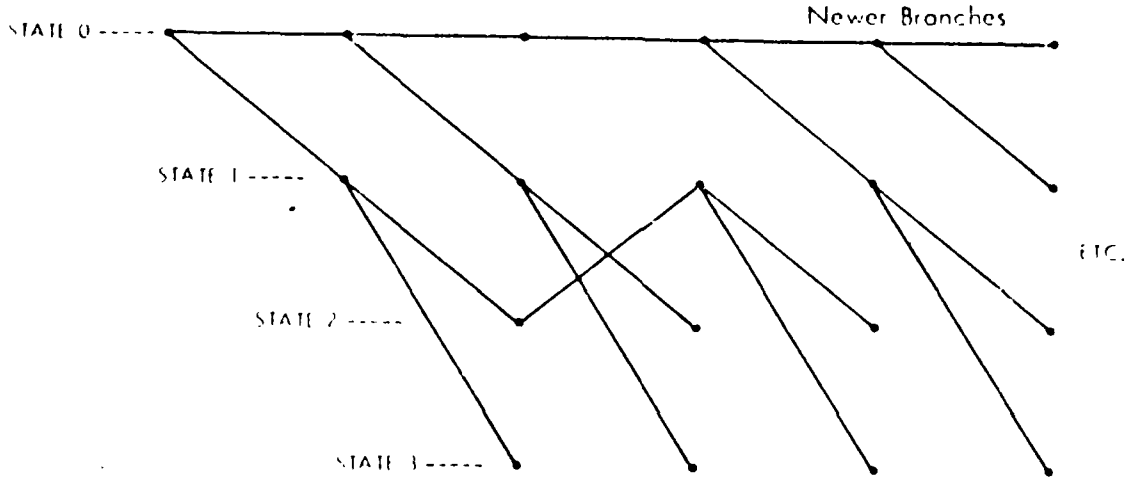


Figure B-1. Example of an Incomplete Trellis Diagram for k=3

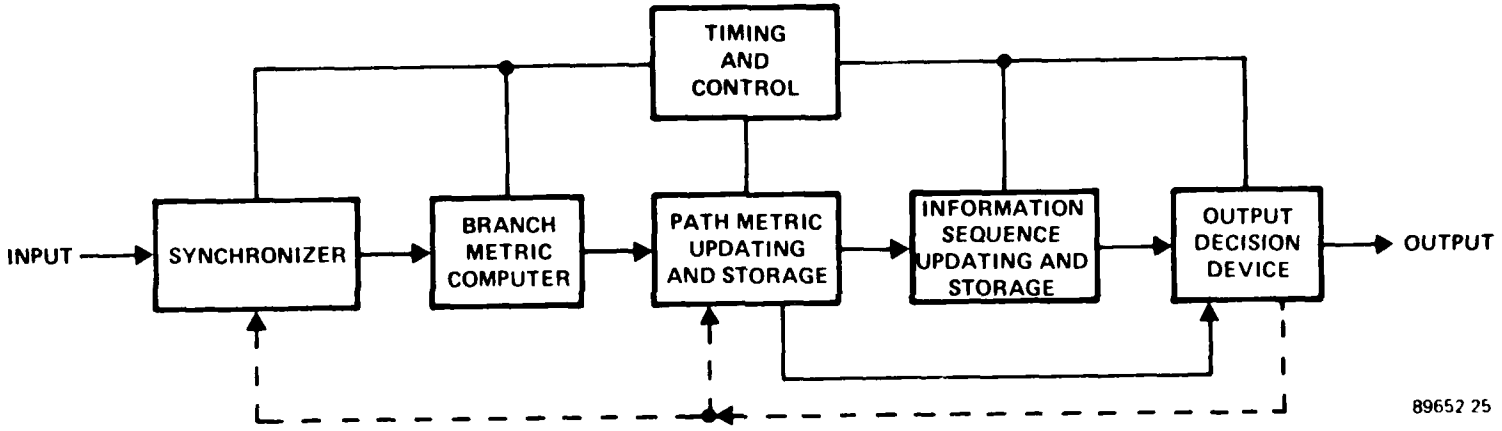


Figure B-2. Functional Block Diagram for a Maximum Likelihood Decoder

branch metrics for each received code branch, (3) a path metric updating, comparison and storage device, in which the branch metrics are added to the previously stored path metrics, the appropriate comparis. .r made and the new path metrics stored, (4) a device for updating and storing the hypothesized information sequences, and (5) an output decision device which determines the decoder output and also provides information to the synchronizer. Each of these devices is discussed in turn in the remainder of this section and the design tradeoffs are elaborated upon.

In general a particular block in this diagram can be implemented using parallel or serial logic depending on the operating speed required and the logic line that is being used. Using fully parallel TTL logic a practical upper limit on the input information rate is on the order of 7 to 10 Mbits/sec. Much higher information rates can be handled using emitter coupled logic; however, the number of integrated circuit packages required is much larger than the TTL machine due to the lack of a variety of Medium Scale Integration packages. This situation is changing rapidly with the introduction of new logic lines, and thus it is undoubtedly true that the practical upper frequency limit will be pushed upward.

Decoder Input - The decoder input is provided in the form of an m-bit binary number by the bit detector. The bit detector would normally consist of a matched filter (or suitable approximation) followed by an A/D converter.

There are three interrelated parameters which need to be determined. These are the number of quantization levels, the spacing between quantization levels, and the number of bits that will be used to represent the branch metric. A particularly simple ad hoc scheme is to use 3 bit (8 level quantization) with equally spaced levels, and a uniform 3 bit metric associated with a single channel bit. That is, the metric would be allowed to take on the values 0 through 7 depending on how closely the received channel bit matched with the hypothesized channel bit. Computer simulations and union bounding techniques have also verified that the quantizing range is not critical and that the probability of decoded bit error exhibits a broad minimum as a function of the spacing between levels with a near optimum setting occurring when the highest slicing level in the A/D converter is chosen to be 1/2 a quantization level below the expected value of the received signal.

Synchronizer - The synchronizer is simply a device for inserting commas (determining the beginning and ending of a branch) in the received bit stream and resolving bit polarity when required. Inserting commas is easily done because if it is done incorrectly the decoder can detect that the sequence is quite different from a legitimate codeword. In general convolutional codes have the property that bit polarity can be resolved without additional information being added to the transmitted bit stream provided a non-transparent code is selected. In this case the sequence resulting after a polarity inversion is not a codeword and again looks like very noisy data to the decoder. If the generator polynomials all have odd weight then the code is said to be transparent to phase inversions. In this case an inversion in the polarity of the decoder input simply produces another codeword which corresponds to an inversion in the decoder output. When this type of code is used, differential encoding/decoding is used outside of the convolutional encoder/decoder to provide the proper polarity for decoded data.

The operation of the synchronizer is straightforward and simply requires an indication that the input sequence contains a much larger number of errors than normal. Several techniques are possible and almost any way that one can get an error detecting indication will work. One method for which there are many possible variations is to look at the rate at which the path metrics are increasing. A second method and one that is particularly simple to implement is to check the oldest bit in each of the hypothesized bit streams and determine whether or not all of the bits are unanimous. In the absence of excessive channel noise the number of non-unanimous decisions is very small whereas with a bit slip or polarity inversion the effective channel noise is sufficiently large that non-unanimous decision will occur a large percentage of the time. Consequently one need only count the number of unanimous decisions that occur in some fixed block length and compare the value with a threshold.

Branch Metric Computer - For rate m/n codes, 2^n different branch metrics need to be computed each branch time. If the uniform assignment method discussed previously is used then the computation is relatively straightforward. For example if the convention is adopted that the binary number 000 corresponds to a highly reliable received "0" and the numbers 001, 010, etc., indicate received zeros with decreasing reliability then the branch metric computer

simply inverts this number to obtain the channel-bit metric associated with a hypothesized "0" or does nothing to it to obtain the channel bit metric associated with a hypothesized "1". To compute the branch metric it need only sum the individual bit metrics that make up the branch. However, we have found that one can choose a different metric assignment in such a way as to greatly reduce the hardware required in implementing the path metric updating and storage portion of the machine in a high speed decoder without sacrificing much coding gain.

Path Metric Updating and Storage - In this portion of the system the received branch metrics are added to the previously accumulated path metrics, the appropriate path metrics are compared, and the largest value in each comparison is retained. Both this portion of the machine and the portion that stores the hypothesized information sequences account for the greatest portion of the parts that are used so it is in these two areas that the greatest care needs to be taken in design in order to avoid excessively large part counts.

The rate 1/2 decoder has the 2^{k-1} survivor metrics stored at each branch time. A new branch is input to the decoder and it is then necessary to compute the new survivor metrics to be ready for the next received branch. This is done by adding the appropriate branch metrics to the appropriate survivor metrics (determined by the code generators). Each path metric is extended out of each state under the assumption of an information zero and under the assumption of an information one. This results in 2^k metrics, two in each of the 2^{k-1} states, as graphically displayed on a trellis diagram. The two metrics representing paths leading into the same state are compared and the larger is stored as that state's new survivor metric. This function is performed for each of the 2^{k-1} states and 2^{k-1} new survivor metrics therefore result. For each of the 2^{k-1} states, one bit of information, which tells which of the two unique paths leading into each state had the larger metric, is fed to the survivor sequence determining function (to be described in the following subsection). Thus 2^{k-1} bits (called trellis connections) are transferred out of the correlation comparators each information bit time. These trellis connections are the distillation of the noisy received channel data by the decoder into the best possible way to draw the incomplete trellis diagram so that decoded bit decisions can be made.

A practical problem is that of minimizing the number of bits required to represent the survivor metrics and implementing a method for rescaling these

metrics to prevent overflow. Minimizing the number of bits will minimize the complexity of the add-compare-select circuit and the amount of storage required for the survivor metrics. We have developed a proprietary technique at Harris Corporation for performing the add-compare-select function which greatly reduces the complexity required in a high-speed decoder yet only sacrifices 0.3 dB in coding gain for a k=7 code. The performance using this technique was shown previously in Figure A-7.

Storage and Updating of Hypothesized Information Sequences - There are basically two different techniques with which this portion of the machine can be handled. The first method, which is called register exchange, calls for a complete hypothesized information sequence to be stored for each of the 2^{k-1} states. The registers in which these sequences are stored are interconnected in precisely the same fashion as the add-compare-select circuits discussed in the previous paragraph. Each time a new branch is processed by the computer, the registers are interchanged corresponding to which sequences survived the comparison, a new bit is added at one end of each register and the oldest bit in each register is delivered to the output decision device. At high speed this technique requires all of the shifting to be done in parallel and means that each individual bit storage device must be equipped with the necessary gates to accept inputs from one of 2 different places. Since one requires that each hypothesized sequence be about 5 constraint lengths this necessitates an enormous amount of hardware and is normally not practical. At very low speeds one may usually design the decoder such that the sequence can be interchanged in a serial fashion through a single routing device. In this case the technique is practical and the number of IC's depend primarily on the constraint length of the code.

Fortunately there is a second method which is most useful when the paths are stored in random access memory (RAM). This is referred to as the traceback method. Using this technique one does not st re the actual information sequences but instead stores the results of each comparison. After several branches have been processed, one recalls the stored decisions in the reverse order in which they were stored and in effect traces back through the trellis diagram. Instead of outputting a single branch worth of information bits this technique requires that one output several branches at a time. One drawback to this technique is that in high speed fully parallel

machines the decoding delay is 2 to 3 times longer than with register exchange since one cannot trace back until several branches past the required decoding depth of 5 constraint lengths have been processed. Typically one might trace back with a 2-to-1 speed advantage. Thus if L branches is the minimum required decoding depth one must provide storage for 3 L sets of branches in order to provide the necessary buffering. However, this approach is still much simpler than direct storage of the sequences. In addition, in a low speed serial machine enough time is available to trace back each bit time. Then one needs to provide storage only for L branches and there is no penalty in terms of storage or decoding delay.

Output Device - Several techniques are available for providing the decoder output. The best of course is to select the sequence possessing the largest path metric and choosing the oldest bit in this sequence. In a high speed fully parallel machine, however, this requires a considerable amount of logic and may not be cost effective. In general we expect all of the sequences will have "merged" by the time the decoding depth has been reached so that all paths in the trellis collapse into a single path. In this case one could just as well choose an arbitrary sequence and output the oldest bit. A method somewhat in between the two from a performance viewpoint is to make a majority decision over all sequences. Many other combinations are obviously possible but probably not practical. Obviously, some degradation takes place by using one of the suboptimum techniques, however, they may be overcome by extending the decoding length . If for example a decoding depth of say 18 is adequate to achieve a given probability of error using the maximum correlated sequence technique then something on the order of 25 will be required using the "majority vote" technique and 30 using the arbitrary sequence technique. A fourth technique which results in very little degradation is to examine the oldest bits in all sequences whose correlation lies above a threshold and make a majority vote. This technique performs almost as well as selecting the maximum correlated sequence but does not require nearly as much hardware.

REFERENCE

B-1. Viterbi, A. J., "Convolutional Codes and Their Performance in Communication Systems," IEEE Trans. on Communication Technology, Vol. COM-19, October 1971, pp. 751-772.

APPENDIX C

PARETO EXPONENTS FOR THE AWGN CHANNEL WITH ERASURES

In this Appendix, we establish the relationships among the Pareto exponent ρ, the symbol erasure probability δ, the code rate R, and E_b/N_o, for binary signaling over the additive white Gaussian noise (AWGN) channel with independent random symbol erasures.

For a discrete memoryless channel with K input symbols and J output symbols, let

$$E_o(\rho) = -\log_2 \sum_{j=0}^{J-1} \left[\sum_{k=0}^{K-1} Q(k)\, P(j|k)^{1/(1+\rho)} \right]^{1+\rho} \qquad (C-1)$$

where $P(j|k)$ are the channel transition probabilities and $Q(k)$ are the probability assignments on input symbols. $E_o(\rho)$ is often called the zero-rate exponent of the channel[C-1]. The Pareto exponent for sequential decoder operation on such a channel with code rate R is given by the implicit solution of the equation

$$R = \frac{E_o(\rho)}{\rho} \qquad (C-2)$$

$E_o(\rho)$ is a convex function of ρ, with $E_o(0) = 0$.

We are interested here in channels obtained by discretizing the AWGN channel by using binary PSK signaling (K=2, Q(k)=1/2) and either hard (J=2) or soft (J=8) demodulator decisions. For the hard decision case,

$$P(0|1) = P(1|0) = p = Q\left(\sqrt{\frac{2E_s}{N_o}}\right) \qquad (C-3)$$

In this case, (C-1) reduces to

$$E_o(\rho) = \rho - \log_2 \left[(1-p)^{1/(1+\rho)} + p^{1/(1+\rho)} \right]^{1+\rho} \qquad (C-4)$$

The code rate for which $\rho=1$, usually denoted R_{comp}, is given by

$$R_{comp} = \frac{E_o(1)}{1} = 1 - \log_2\left[1 + 2\sqrt{p(1-p)}\right] \qquad (C-5)$$

We can use (C-3), (C-5), and the relation $E_s = RE_b$ to obtain, as a function of R, the value of E_b/N_o necessary to make $R = R_{comp}$. Since $\rho=1$ is considered the marginal operating point for a sequential decoder, the resulting value represents the minimum E_b/N_o required for sequential decoding. This minimum value is shown as the J=2 curve of Figure C-1.

When 8-level decisions are used, the transition probabilities $P(j|k)$ depend on the placement of the decision boundaries. Massey[C-2] and Lee[C-3] have developed methods to find decision region boundaries which maximize R_{comp}. In an earlier paper, Jacobs[C-4] used boundaries at 0, ±1/2, ±1, ±3/2, assuming that the demodulator output is normalized so that the noise has unit variance. Jacobs observed that the resulting values of E_b/N_o required to achieve $R=R_{comp}$ were within 0.2 dB of the value predicted with infinitely fine quantization so that the loss due to using these boundaries is not significant. These results are shown as the J=8 curve of Figure C-1. Note that the use of 8 levels rather than hard decisions yields about 2 dB improvement over the interesting range of code rates.

By proceeding in an analogous fashion, using (C-1), (C-2), and $E_s=RE_b$, the required E_b/N_o to sustain any desired value of ρ as a function of code rate R can be obtained.

Suppose now that in addition to the effects of AWGN, random symbol erasures occur on the channel. An erasure is defined as a received symbol J for which $\tilde{P}(J/0) = \tilde{P}(J/1)$, so that J conveys no information as to the identity of the transmitted symbol. Let δ denote the probability of a symbol erasure. The symbol transition probabilities for symbols other than the erasure are given by

$$\tilde{P}(j|k) = (1-\delta)P(j|k) \quad 0\leq j\leq J-1,\ k=0,1 \qquad (C-6)$$

where $P(j|k)$ is the corresponding transition probability in the absence of erasures. Figure C-2 shows the hard-decision channel with and without erasures.

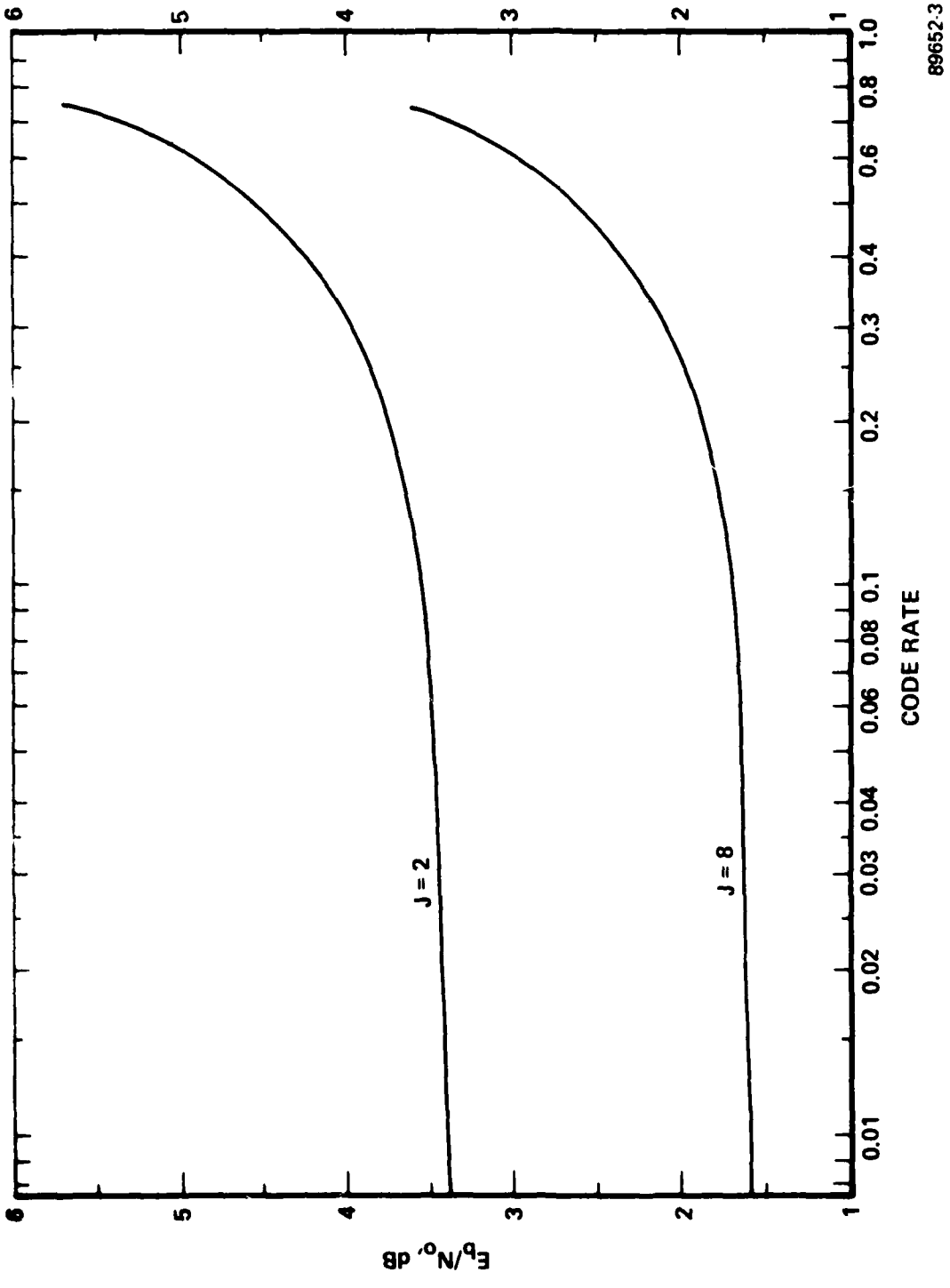


Figure C-1. E_b/N_o Required to Achieve $R=R_{comp}$

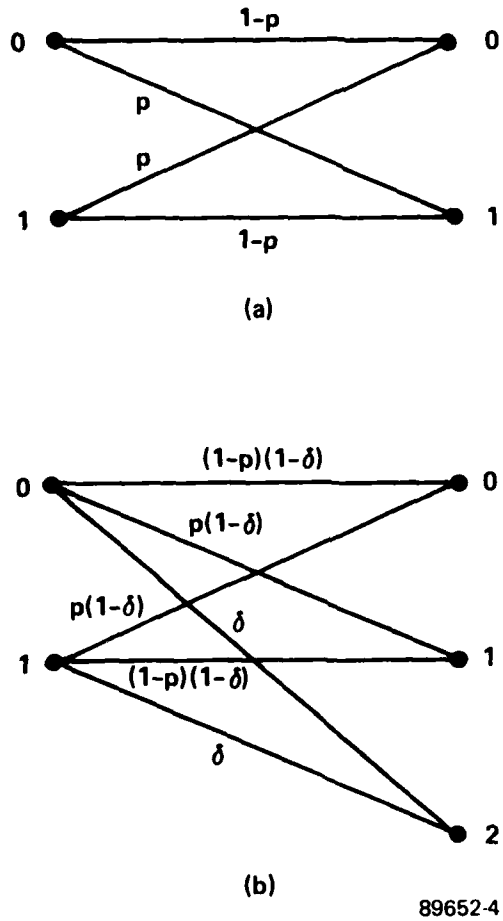


Figure C-2. Hard Decision Channel:
(a) Without Erasures; (b) With Erasures

We will denote the zero-rate exponent of the erasure channel by $\tilde{E}_0(\rho)$. Substituting (C-6) into (C-1), we obtain

$$\tilde{E}_0(\rho) = -\log_2 \{\delta + (1-\delta)2^{-E_0(\rho)}\}$$

where $E_0(\rho)$ is the zero-rate exponent in the absence of erasures. Therefore we have

$$2^{-\tilde{E}_0(\rho)} = \delta + (1-\delta)2^{-E_0(\rho)} \qquad \text{(C-7)}$$

It is of interest to determine the effect of erasures on very quiet channels, i.e., $E_b/N_0 \to \infty$. Under this circumstance, it is clear that nothing is gained by using soft decisions, so we can use (C-4) with $p \to 0$ to obtain the zero rate exponent in the absence of erasures:

$$E_0(\rho) \to \rho \text{ as } p \to 0$$

Then from (C-7) we have

$$2^{-\tilde{E}_0(\rho)} = \delta + (1-\delta)2^{-\rho}$$

or

$$\tilde{E}_0(\rho) = -\log_2 \{\delta + (1-\delta)2^{-\rho}\}$$

Since the Pareto exponent when operating at rate R is the solution of $R=\tilde{E}_0(\rho)/\rho$, we have

$$2^{-\rho R} = \delta + (1-\delta)2^{-\rho} \qquad \text{(C-8)}$$

Or letting $\chi=2^{-\rho}$,

$$\chi^R = \delta + (1-\delta)\chi$$

We are therefore looking for the intersection of the curve χ^R with the straight line $\delta+(1-\delta)\chi$, where $0\leq\chi\leq1$, as illustrated in Figure C-3. Note that the slope of χ^R at $\chi=1$ is R, and that if the slope of the straight line does not exceed R, no such intersection will exist. Therefore the rate must satisfy

$$R \leq 1-\delta \qquad (C-9)$$

The presence of erasures places a limit on the achievable Pareto exponent and operating rate regardless of the signal-to-noise ratio in the intervals between erasures. The best exponent in the absence of channel noise is given by the solution of (C-8). This exponent is an upper bound to achievable ρ for given R and δ, regardless of E_b/N_o. Best achievable exponents for selected rates are shown as functions of δ in Figure C-4. In accordance with (C-9), $\rho=0$ is achieved at $\delta=1-R$.

Since the sequential decoder computational cutoff rate is given by

$$R_{comp} = E_o(1)$$

we have from (C-7)

$$2^{-\tilde{R}_{comp}} = \delta + (1-\delta)\, 2^{-R_{comp}}$$

For the hard decision channel, we can insert (C-5) into this equation to obtain

$$\tilde{R}_{comp} = 1-\log_2\{1+\delta+2(1-\delta)\sqrt{p(1-p)}\} \qquad (C-10)$$

For operation at $R=\tilde{R}_{comp}$, (C-8) implies a reduction in p (or increase in E_b/N_o) with increasing δ to satisfy the equality. A similar situation holds for the case of eight-level quantization. Moreover, an analogous determination of required E_b/N_o to achieve any specified ρ in the presence of erasures at probability δ can be made using (C-2), (C-7), and the channel transition probabilities as determined by R, E_b/N_o, and the type of output quantization. Figures C-5 and C-6 show the required E_b/N_o to achieve $\rho=2$ for hard and soft decisions, respectively. In the latter case, the decision regions are those used by Jacobs[C-4].

Finally, to demonstrate the effect of output quantization, Figure C-7 illustrates the required E_b/N_o as a function of ρ for hard decisions, 8-level

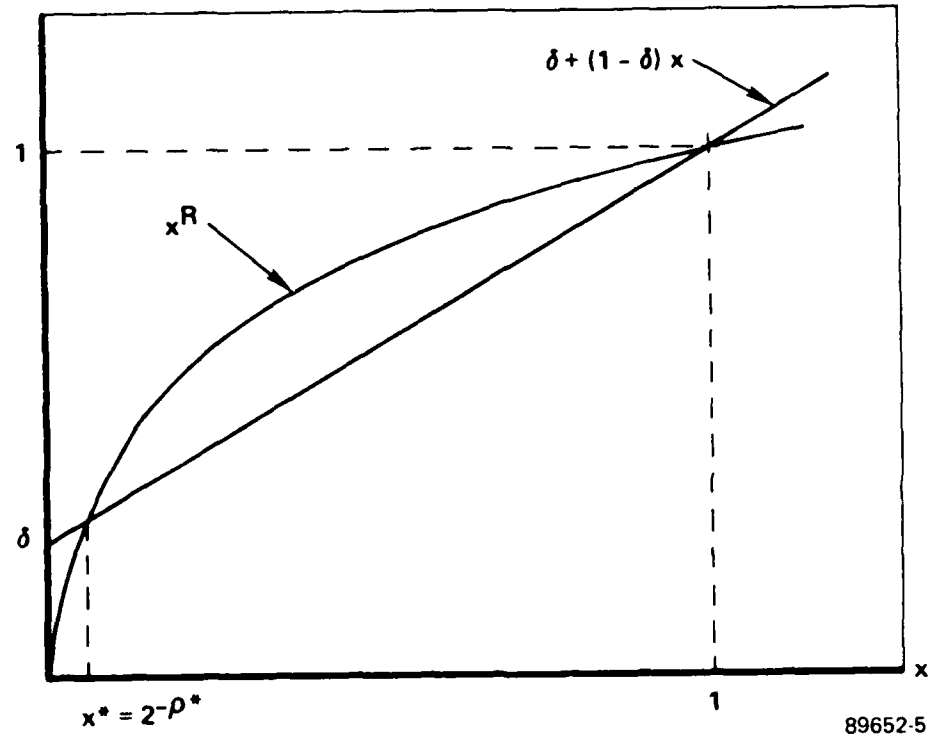


Figure C-3. Determination of Maximum Achievable Pareto Exponent at Rate R With Erasure Probability δ

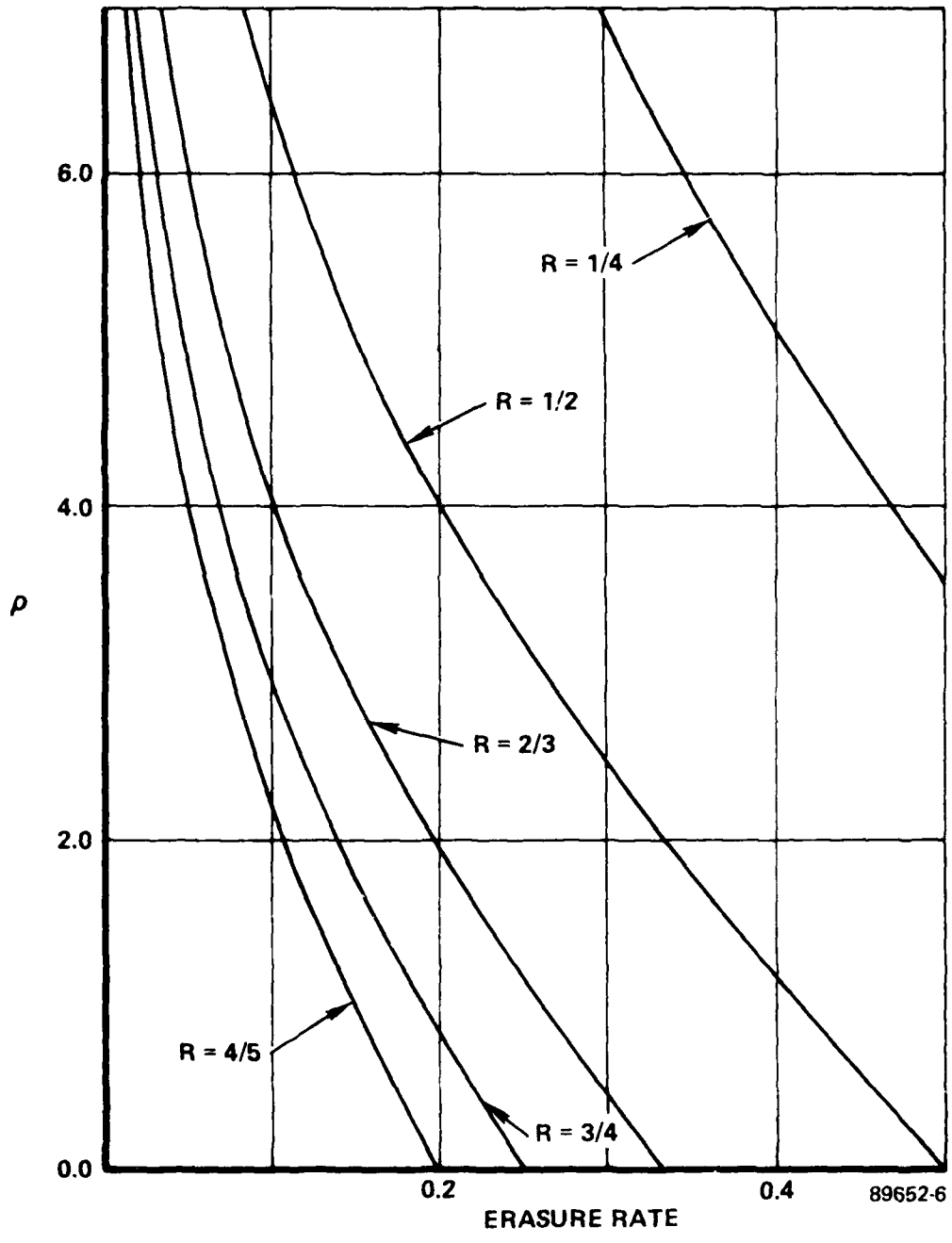


Figure C-4. Maximum Achievable Pareto Exponent

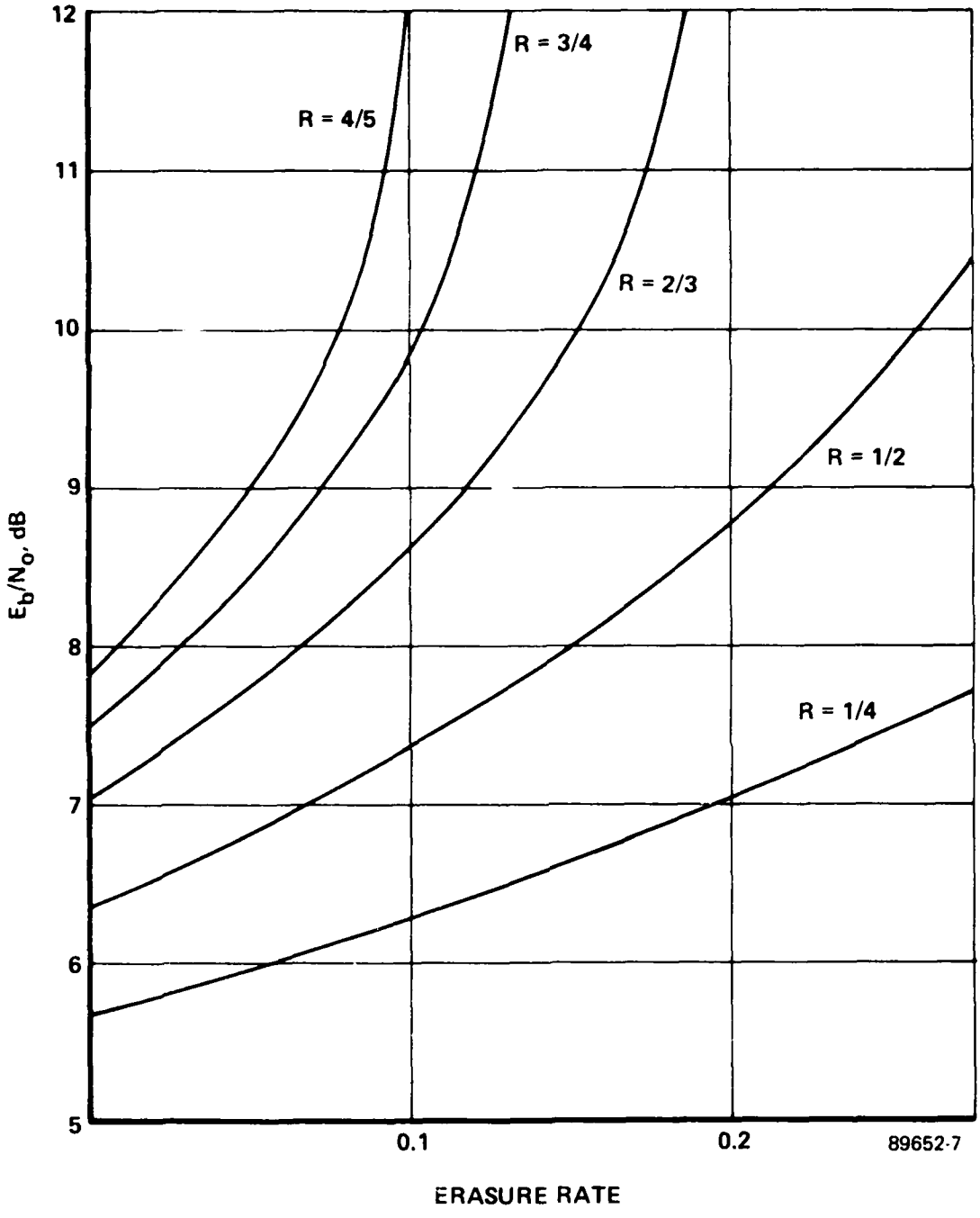


Figure C-5. E_b/N_o Required to Achieve $\rho=2$, Hard Decisions

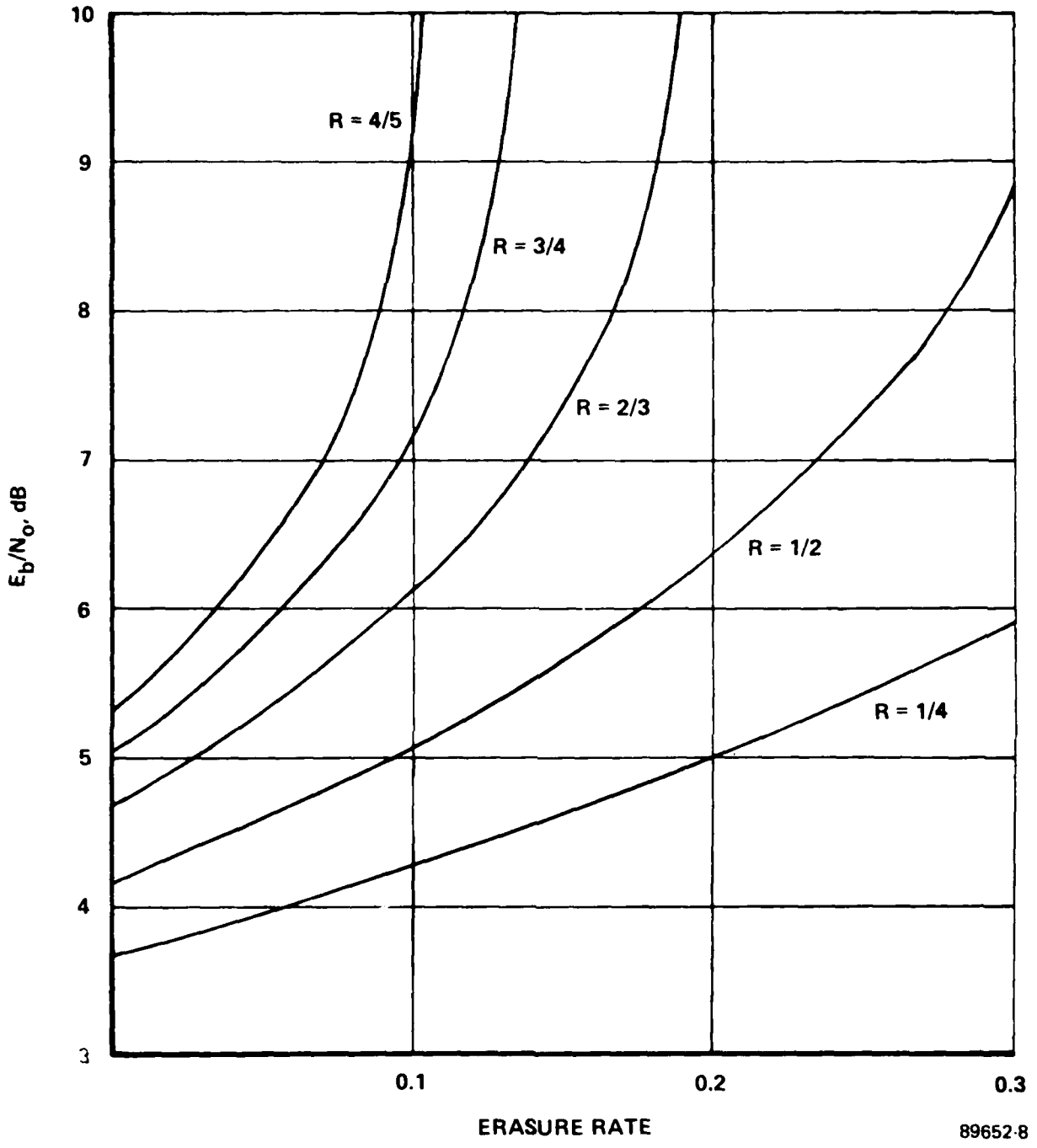


Figure C-6. E_b/N_0 Required to Achieve $\rho=2$, Soft Decisions

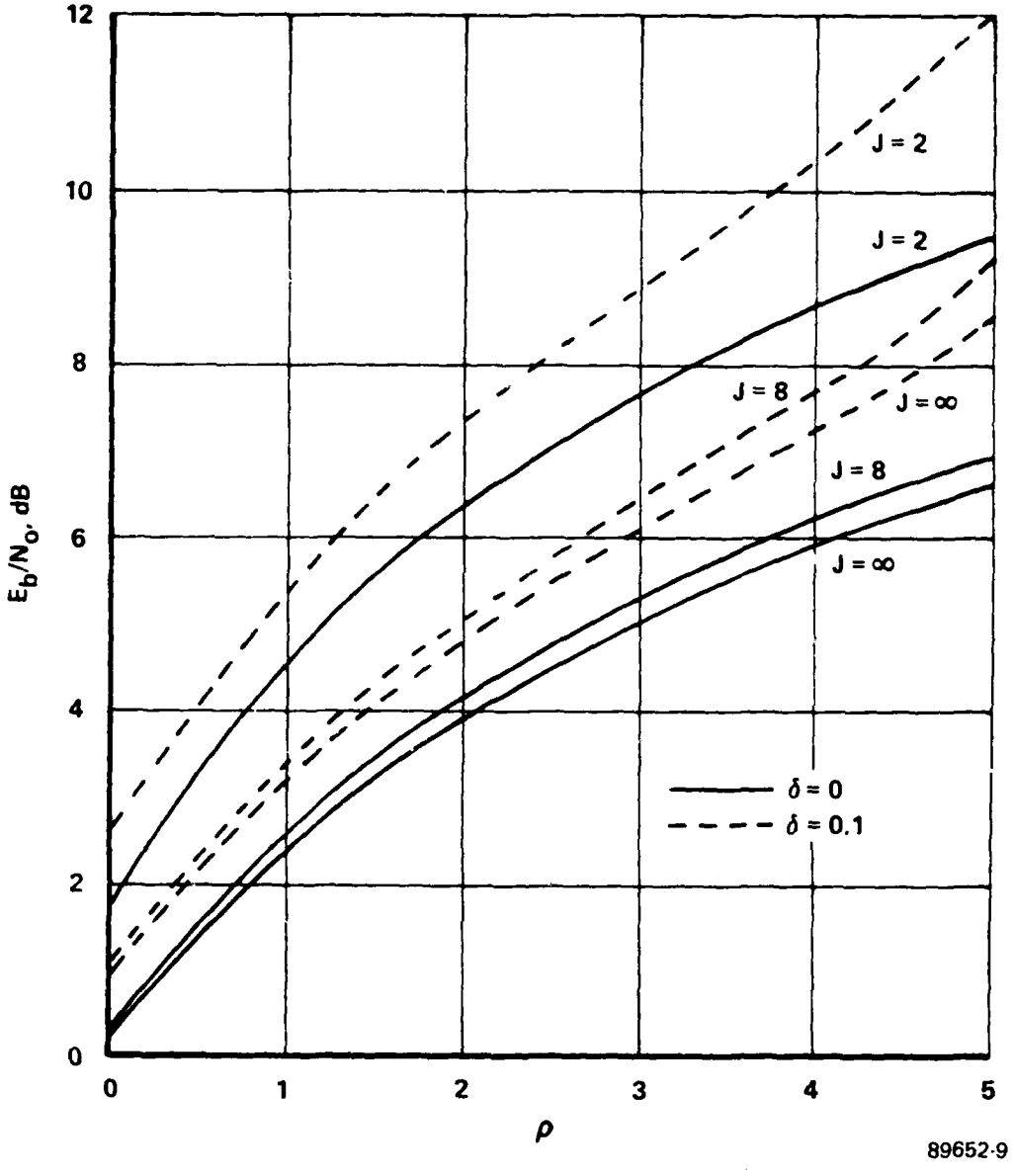


Figure C-7. E_b/N_0 Required to Achieve Pareto Exponent ρ, R=1/2

(Jacobs) decisions, and unquantized channel output. For simplicity only R=1/2, is considered, although results are given for both no erasures and 10% erasures.

REFERENCES

C-1. R. G. Gallager, Information Theory and Reliable Communication, Wiley, 1965.

C-2. J. L. Massey, "Coding and Modulation in Digital Communications," Proc. Int. Zurich Seminar on Digital Comm., March 1974.

C-3. L. N. Lee, "On Optimal Soft-Decision Demodulation," IEEE Trans. Information Theory, Vol. IT-22, pp. 437-444, July 1976.

C-4 I. M. Jacobs, "Sequential Decoding for Efficient Communication from Deep Space," IEEE Trans. Comm. Tech., Vol. COM-15, pp. 492-501, August 1967.

APPENDIX D

PERFORMANCE OF REED-SOLOMON CODES

In this Appendix we evaluate the performance of Reed-Solomon codes when used to correct bit errors due to thermal noise and periodic erasures caused by pulsed RFI. The pulsed RFI is blanked prior to the demodulator and causes periodic bursts of erased bits. As a design goal, erasure rates of up to 10% should be accommodated. We determine the amount of coding gain achievable (relative to uncoded PSK at an error rate of 10^{-5} in the absence of erasures) under these conditions for three techniques employing Reed-Solomon (RS) codes. As described in Paragraph 3.2.2, these techniques are:

- Conventional RS encoding and decoding without interleaving.
- RS encoding with random interleaving on a symbol-by-symbol* basis.
- Concatenation of an RS outer code with an overall parity check as an inner code, with interleaving on a bit-by-bit basis.

Let δ denote the raw channel erasure rate, i.e., the duty cycle of the blanking. As a minimum, this erasure rate will cause the same fraction of RS code symbols to be erased. However, if the erasures caused by the pulse interference are not aligned exactly with the RS code symbols, erasure amplification results and the fraction of symbols erased is greater than δ. Denoting the pulse length and symbol length (in seconds) by τ and τ_s, respectively, the interference period in symbols is

$$T_s = \frac{\tau/\delta}{\tau_s}$$

A symbol is considered erased if any of the bits in its representation as a binary m-tuple are erased. The average number of symbols erased per pulse is $1+\varepsilon$, where $\varepsilon=\tau/\tau_s$. Thus the average symbol erasure rate, provided that $\varepsilon>\delta$, is

*The term "symbol" as used in this Appendix refers to an element of $GF(2^m)$, which will be represented as a binary m-tuple.

$$\Delta_{AVG} = \delta\left(\frac{1+\varepsilon}{\varepsilon}\right) \qquad \text{(D-1)}$$

(If $\delta \geq \varepsilon$, erasure pulses occur at least once per symbol so that all symbols are erased and $\Delta_{AVG}=1$.) Equation (D-1) indicates that the erasure rate apparent to the decoder is greater, on the average, than that on the raw channel. Henceforth, we will use the symbol Δ to indicate erasure rates of symbols to distinguish these rates from the raw channel erasure rate of δ.

Depending upon the relationship between the code length, the interference period, and the ratio τ/τ_s, there can be considerable variation in the number of erasures per codeword, and thus additional erasure amplification on some codewords. An upper bound on the erasure rate in any single codeword can be calculated in the following manner. The worst case occurs when the codeword contains exactly h periods of the interference plus the entire pulse from the next period of the interference. Then the code block length in symbols can be written

$$n = h\,T_s + \lceil \varepsilon \rceil,$$

where $\lceil x \rceil$ denotes the least integer not less than x. The codeword contains h+1 interfering pulses and each pulse can erase at most $\lceil \varepsilon \rceil + 1$ symbols. Then the erasure rate is upper bounded by

$$\Delta_{max} = \frac{(h+1)(\lceil \varepsilon \rceil + 1)}{h\,\varepsilon + \delta\,\lceil \varepsilon \rceil}$$

$$= K(\delta,\varepsilon,h)\,\delta$$

where $K(\delta,\varepsilon,h)$ represents an upper bound on the erasure amplification factor. Since the worst-case erasure amplification must be at least as great as the average amplification, (D-1) provides a lower bound of $(1+\varepsilon)/\varepsilon$ for the worst-case amplification. These bounds are shown in Figure D-1 for h=5 and δ=0.05, which are representative values for our problem. Moreover, there is little sensitivity of $K(\delta,\varepsilon,h)$ to changes in the values of h and δ. Increasing δ only slightly decreases $K(\delta,\varepsilon,h)$ as long as $\delta \ll h$. Increasing h also causes only a slight decrease in $K(\delta,\varepsilon,h)$. At lower values of h an increase in $K(\delta,\varepsilon,h)$ is observed, but it only becomes significant for h<2. Accordingly, we can use

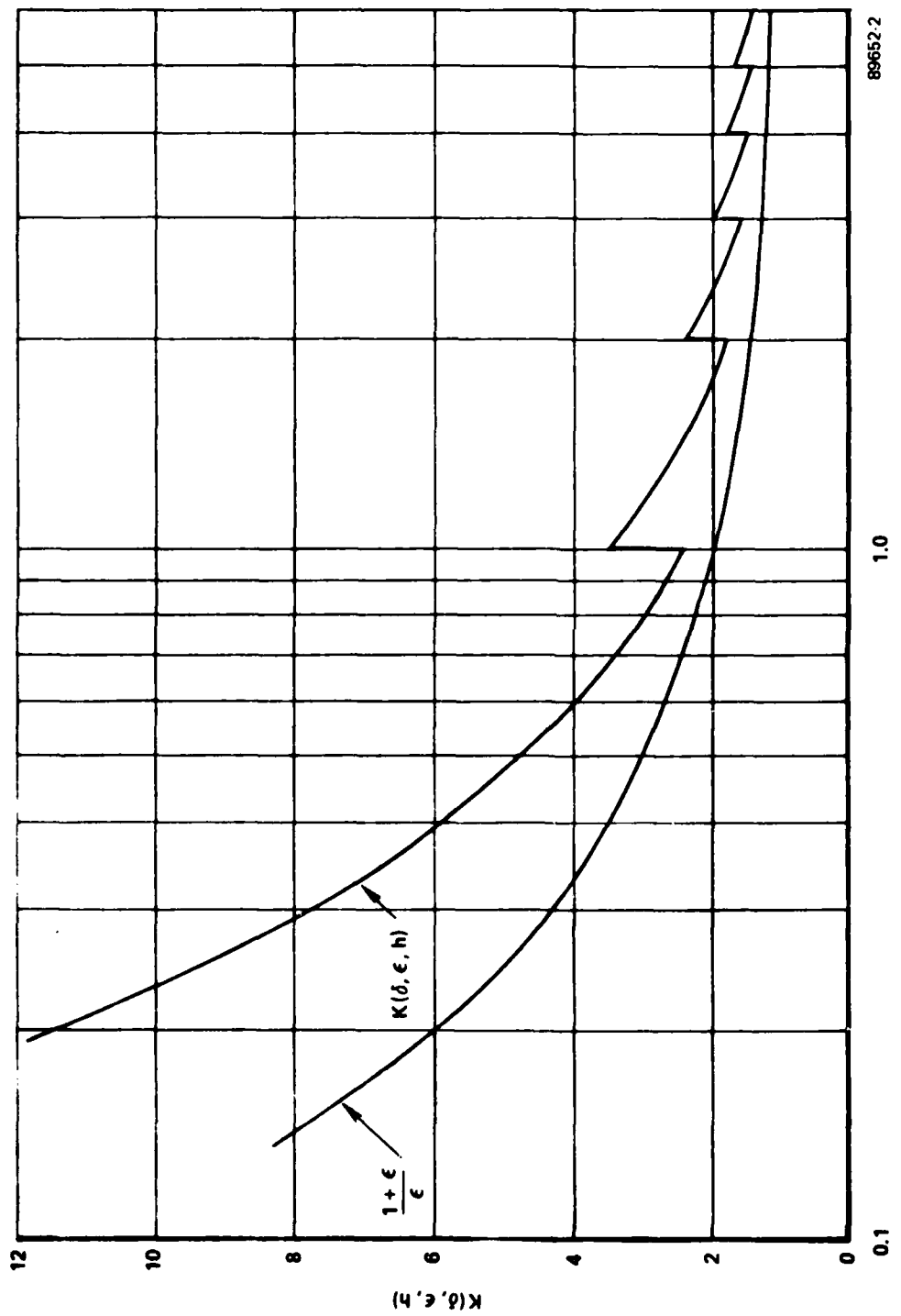


Figure D-1. Erasure Amplification for h=5, δ=0.05

Figure D-1 to predict erasure amplification with reasonable accuracy for all δ and h of interest.

At higher values of ε (many symbols per pulse) the upper and lower bounds are rather close together, and the resulting erasure amplification is small. For example, if the interfering pulse is 8 symbols long (ε=8) the erasure amplification factor is in the range 1.35 to 1.5. Thus, a duty cycle of 0.05 causes a fraction of erased symbols of at most 0.075 (assuming h=5) which can be corrected with a redundancy of 0.075 (not allowing for correction of channel errors). The RS code is most efficient in correcting the pulse interference for parameters in this range, i.e., that there be several periods of interference within the block (h>2) and that the pulse length be considerably greater than the symbol length (ε large). If one were confident that over the expected range of interference parameters that ε and h would both be large, then the RS code would not need to be interleaved. The reason for this, of course, is that if one uses a pseudo-random interleaving approach the average erasure amplification factor could be reduced to the lower bound in Figure D-1 which is $(1+\varepsilon)/\varepsilon$. However, the actual number of erasures in a given codeword will vary above and below this amount. Since the worst-case for the non-interleaved code is so near the average erasure amplification factor, there is little if anything to be gained by interleaving.

Unfortunately, these parameters are not in this range for our problem. Since FLTSATCOM channels have a bandwidth of 25 kHz, the upper limit on transmission rates will be a few tens of kilobits, say 20 kbps nominal. The most troublesome RFI source is periodic with a duty cycle of 0.05 and a PRF in the range 230-250. Using the upper limit on PRF the pulse length is just τ=200 μsec, which is 4 bits at a 20 kbps rate. The practical range of RS code symbol lengths is 4 to 8 bits, resulting in values of ε in the range 0.5 to 1.0. There is considerable erasure amplification for values of ε in this range. The situation could get better only if transmission rates were much higher (which is not likely to occur with these 25 kHz channels without much more power). For lower data rates, the erasure amplification gets much worse.

The performance of the non-interleaved RS code can be calculated in a straightforward manner as a function of the fraction of erased symbols, Δ. Assume that the RS code has length n symbols with k information symbols per block. The minimum distance between the transmitted codeword and all other

codewords is reduced by no more than the number of symbol erasures that occur. The distance between each transmitted codeword and all other codewords is no less than

$$d^* = d - \lceil \Delta n \rceil$$

$$= n - k + 1 - \lceil \Delta n \rceil$$

$$= \lfloor (1-\Delta) n \rfloor - k + 1$$

$$= n^* - k + 1 \quad ,$$

where $\lfloor x \rfloor$ is the greatest integer not exceeding x. Assuming d^* is positive, the number of symbol errors that can be corrected in addition to the erasures is

$$t^* = \lfloor (d^*-1)/2 \rfloor \ .$$

If a word error occurs then no more than $(d-d^*+t^*+i)$ symbols errors can be made in decoding the n symbols (for d^* odd), where i is the number of received symbol errors. This can also be written as $(d-\lfloor (d^*+1)/2 \rfloor+i)$ which is valid for all d^*. We assume that each decoded symbol error has half its bits in error. Then the probability of bit error can be estimated by union bounding as

$$P_b = \frac{1}{2n} \sum_{i=t^*+1}^{n^*} (d-\lfloor (d^*+1)/2 \rfloor+i) \binom{n^*}{i} p_s^{\,i} (1-p_s)^{n^*-i},$$

where p_s is the raw symbol error rate. If p is the bit error probability on the raw channel, the probability of an m-bit symbol error is

$$p_s = 1-(1-p)^m,$$

These relations have been used to determine the coding gain using RS coding for several values of symbol erasure rate Δ and code rate R for code lengths of 31 and 63 symbols. These results are tabulated in Table D-1. The coding

Table D-1.

Achievable Coding Gain for Selected RS Codes
(All Gain Values in Decibels)

			Symbol Erasure Rate (Δ)			
n	k	R	0.05	0.10	0.15	0.20
31	8	0.258	0.4	0.7	0.1	0.0
31	10	0.323	1.1	0.8	0.8	0.3
31	15	0.484	2.0	1.7	1.3	0.8
31	21	0.677	2.1	1.5	0.7	-0.5
31	23	0.742	1.9	1.1	-0.2	
31	25	0.806	1.5	0.3		
63	16	0.254	1.0	0.9	0.5	0.4
63	21	0.333	1.9	1.6	1.6	1.3
63	32	0.508	2.6	2.5	2.1	1.9
63	42	0.667	2.7	2.4	1.8	1.3
63	47	0.746	2.5	1.8	1.2	0.1
63	51	0.810	2.1	0.9	-0.1	

gain is measured relative to uncoded PSK at $P_b=10^{-5}$ with no interference. These results show that for the erasure rates shown, the peak coding gains are for code rates in the range $0.4<R<0.6$. For higher rates the gain declines because there is not enough redundancy to correct the erasures, while at lower code rates the gain declines because the additional redundancy is not used efficiently enough to make up for the bandspreading loss.

The erasure rate Δ listed as a parameter in Table D-1 is the symbol erasure rate which is considerably higher than the RFI duty cycle δ because of erasure amplification. We can estimate the impact of this on the coding gain of a non-interleaved RS code as follows. Again assuming 20 kbps transmission rate, and a single 250 Hz PRF with $\delta=0.05$, a 5-bit symbol results in a value of $\varepsilon=0.8$. From Figure D-1, worst-case erasure amplification is 3, so that $\Delta=0.15$. From Table D-1, we find that the coding gain at $P_b=10^{-5}$ is -0.2 dB for R=0.74 and 0.74 dB for R=0.68 (we are assuming performance to be dominated by worst-case symbol erasure rate). If one tries the larger block length, n=63, then ε drops to 0.67 and the symbol erasure rate increases to 0.175. This value of Δ is not tabulated, but by interpolating between $\Delta=0.15$ and $\Delta=0.20$, we conclude that the gain is increased by a little more than 0.5 dB. The considerable amount of erasure amplification present for both codes costs about 1 to 2 dB depending upon the rate. Furthermore, if two sources of RFI are present the worst-case symbol erasure rate will become 0.30 for n=31 and 0.35 for n=63. Obviously, code rates less than 0.7 must be used. In fact, with n=63, R=0.51 will give less than 1.0 dB of gain. Thus, with RS coding at a 20 kbps rate, a moderate coding gain can be obtained against one RFI source for codes rates in the neighborhood of 2/3 to 3/4. However, this gain disappears when two RFI sources are present. One must reduce the code rate to 1/2 to achieve a moderate gain. For lower data rates, say at 10 kbps, this coding approach performs even worse since the erasure amplification factor is increased by about a factor of two. Obviously, one can use more redundancy to fight this problem, but it must be used in a sensible manner.

The second approach evaluated is the use of pseudo-random interleaving of the RS code symbols. The interleaver period is chosen to be several codewords to make the average symbol erasure rate per codeword to be equal to Δ_{AVG} given by (D-1) and be nearly binomially distributed. Then one could use the RS code to perform correction of both erasures and errors. The symbol error [illegible] is given by

$$p_s = [1-(1-p)^m](1-\Delta_{AVG}),$$

where p is the raw BER assuming no interference. The symbol erasure rate is Δ_{AVG}.

Performance can be evaluated as follows. Let e denote the number of erasures that occur in a given codeword. An RS code can correct all combinations of errors and erasures such that 2t+e<d. Then the word error rate using this technique is bounded by

$$P_w \le \sum_{i=0}^{n} \sum_{\substack{e=d-2i \\ e\ge 0}}^{n-i} \binom{n}{i,e} p_s^{\,i} \Delta_{AVG}^{\,e} (1-p_s-\Delta_{AVG})^{n-e-i} ,$$

where

$$\binom{n}{i,e} = \frac{N!}{i!e!(n-i-e)!}$$

The RS decoder operating in this mode will never correct more than $\lfloor (d-1-e)/2 \rfloor$ errors. Thus, the bit error rate is bounded by

$$P_B \le \sum_{i=0}^{n} \sum_{e=d-2i}^{n-i} \frac{n(e,i)}{2n} \binom{n}{i,e} p_s^{\,i} \Delta_{AVG}^{\,e} (1-p_s-\Delta_{AVG})^{n-e-i}, \qquad (D-2)$$

where

$$n(e,i) = e+i+\max [0, \lfloor (d-1-e)/2 \rfloor] .$$

The coding gain of RS codes using pseudo-random symbol interleaving with code block lengths of 31 and 63 for several code rates was evaluated using (D-2). The results are tabulated in Table D-2. In comparing with Table D-1 we note that for a specific value of Δ the coding gain is typically less with interleaving, particularly at higher rates. The reason is, of course, that the number of erasures per codeword is fixed for the first approach, while the number of erasures per codeword is a random variable with average value $n\Delta_{AVG}$ for the second approach.

Table D-2.

Achievable Coding Gain for Selected Interleaved RS Codes
(All Gain Values in Decibels)

			Average Erasure Rate (Δ_{AVG})			
n	k	R	0.05	0.10	0.15	0.20
31	11	0.355	1.2	1.1	0.6	0.3
31	16	0.516	2.0	1.4	0.5	
31	21	0.677	1.6			
63	20	0.317		1.7	1.3	0.8
63	32	0.508	2.8	2.6	1.8	1.1
63	40	0.635	2.9	1.7	1.3	
63	48	0.762	2.1			
63	50	0.794	1.7			

To repeat the previous example for the interleaved RS approach, a 5 bit symbol (and a 31 symbol codeword) results in ε=0.8. Then Δ_{AVG}=0.11 for one RFI source (δ=0.05). From Table D-2, the coding gain at R=0.52 is about 1.3 dB. There is no coding gain available at rates in the range of 2/3 to 3/4. With the 6 bit symbol and a block length of 63 symbols we have ε=0.67, and Δ_{AVG}=0.125 for one RFI source. In this case the gain available is about 2.0 dB for R=0.51 and 1.5 dB for R=0.65. The situation is much worse for two RFI sources (δ=0.10). For the 5 bit symbol Δ_{AVG}=0.22 and for the 6 bit symbol Δ_{AVG}=0.25. In both cases there is no coding gain available at rates of 1/2 and greater. We have not extensively evaluated longer codes because the situation is not promising for achieving the higher code rates due to erasure amplification. For example, if m=8, then ε=0.5 and δ=0.10 would result in Δ_{AVG}=0.30. Obviously, no code could achieve a rate greater than $(1-\Delta_{AVG})$=0.7, and in fact a moderate amount of gain with the RS code could only be achieved at rates considerably less than that.

The last approach examined was a concatenated code approach using an overall parity check on each symbol giving an (m+1,m) distance-2 code. Erasures will be pseudo-randomly bit-interleaved over a sufficiently long span to make the average bit erasure rate seen by any codeword be very close to δ and be binomially distributed. Then the concatenated code will be used as a random erasure and error correcting code. The code rate is R=m k/n(m+1). The inner code will be used to correct single erasures. However, if two or more bit erasures occur in any symbol, the symbol will be considered to be erased. The symbol erasure rate due to RFI with duty cycle δ is then

$$p_e(\delta) = 1 - (1-\delta)^{m+1} - (m+1)(1-\delta)^m\delta. \qquad \text{(D-3)}$$

If hard decisions on each bit were used, one could calculate the symbol error rate and the contribution to symbol erasure rate due to detectable but uncorrectable single bit errors and use these expressions with (D-2) to compute performance. However, the best approach would be to use soft decisions to allow the symbol error rate to be significantly reduced while allowing an increase in the symbol erasure rate. Unfortunately, a simple lower bound on performance will show that even this technique will not provide a moderate coding gain at higher code rates when δ=0.10.

With 8-level quantized bit decisions, an efficient set of quantization thresholds for binary antipodal signalling with signals at $\pm\sqrt{E_N}$ is the set of levels

$$\frac{1}{7}(-8+2i)\sqrt{E_N},\ i=1,2,\ldots,7.$$

We assume a bit erasure occurs if the matched filter output voltage falls in one of the middle two quantization intervals, i.e., in the range $(-\frac{2}{7}\sqrt{E_N},+\frac{2}{7}\sqrt{E_N})$. This can be caused either by blanking or by thermal noise, which leads to a composite bit erasure rate of

$$\delta_c = \delta + (1-\delta)\left[Q\left(\frac{5}{7}\sqrt{\frac{2E_N}{N_o}}\right) - Q\left(\frac{9}{7}\sqrt{\frac{2E_N}{N_o}}\right)\right]. \qquad \text{(D-4)}$$

The composite bit erasure rate, δ_c, can be substituted into (D-3) to determine the symbol erasure rate. The symbol erasure rate for m=5 and 6 and δ=0.05 and 0.10 as a function of E_N/N_o is shown in Figure D-2.

Since the parity bit added to each m-bit symbol results in considerable bandspreading, an obvious question one might ask is how this affects the maximum achievable code rate. An upper bound on the achievable code rate for this concatenation scheme can be developed as follows. The RS code must correct symbol erasures at a rate of $p_e(\delta_c)$ as given by (D-3) and (D-4). Thus, the RS code rate must be less than $(1-p_e(\delta_c))$. Since the inner code has rate m/(m+1), the overall rate of the concatenated code must be less than

$$R_{max} = \frac{m}{m+1}\left[1-p_e(\delta_c)\right]. \qquad \text{(D-5)}$$

The quantity $p_e(\delta_c)$ depends on E_N/N_o. For operation at code rate R_{max}, E_N/N_o is given by

$$\frac{E_N}{N_o} = R_{max}\frac{E_b}{N_o} \qquad \text{(D-6)}$$

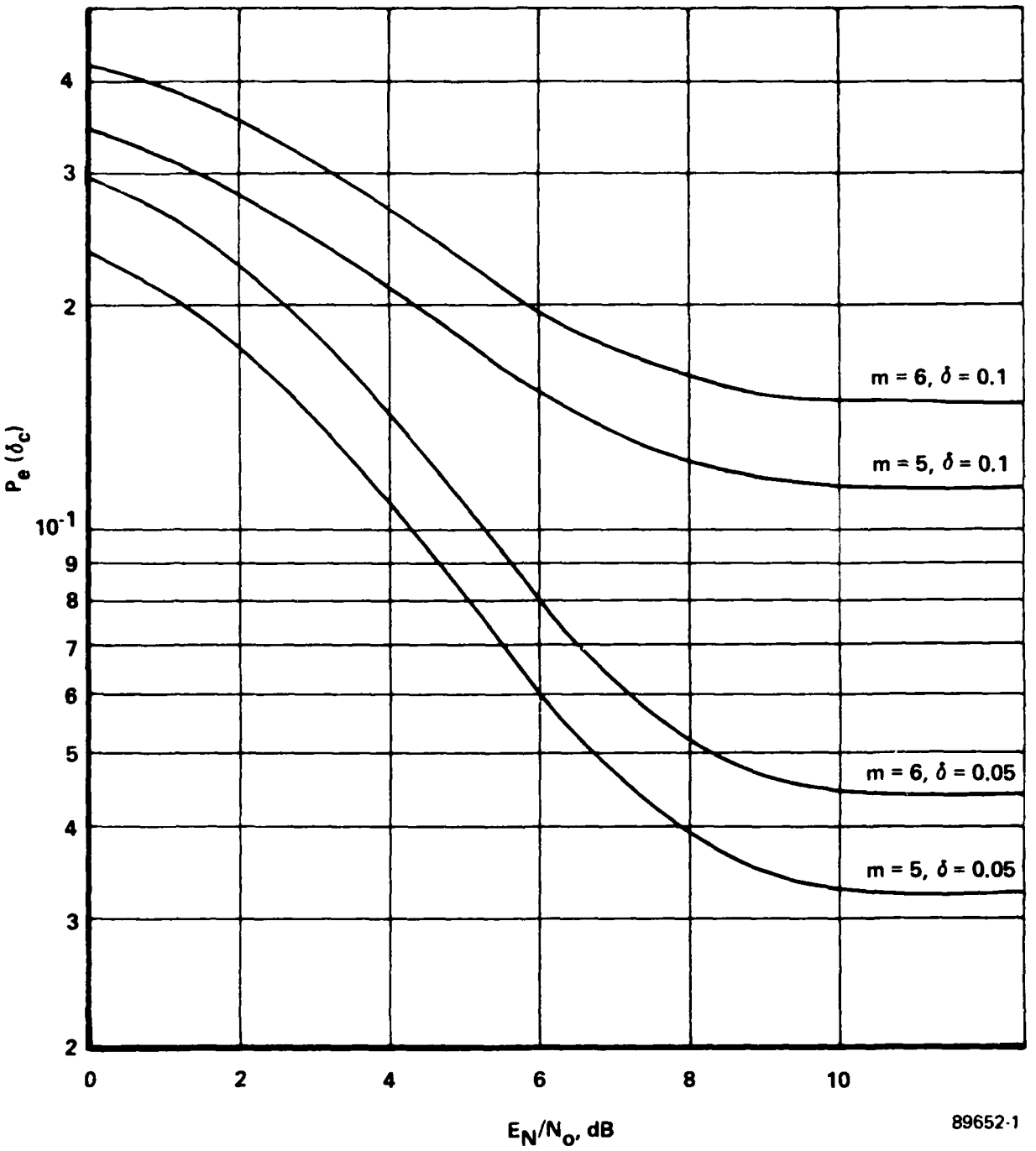


Figure D-2. Composite Symbol Erasure Rate

To achieve any specified coding gain, E_b/N_o is given in decibels by

$$\left(\frac{E_b}{N_o}\right)_{dB} = 9.6 - (\text{Gain})_{dB} \qquad \text{(D-7)}$$

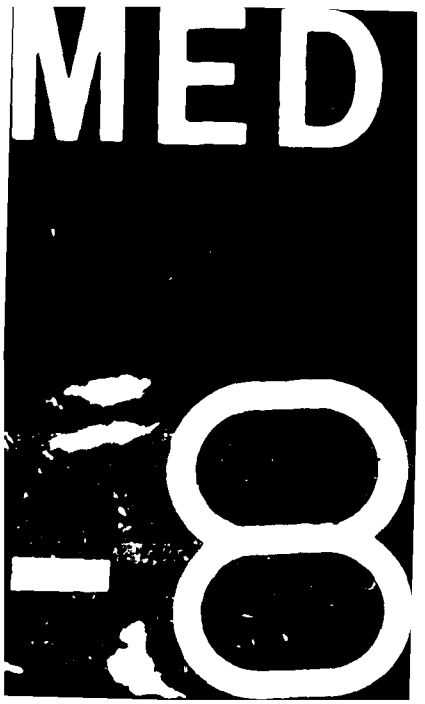
Thus the desired coding gain, the underlying erasure rate δ, and the RS symbol length m determine, by way of equations (D-3) through (D-7), the maximum possible code rate R_{max}. For 1 dB of coding gain and δ=0.10, maximum rates are listed for various symbol lengths in Table D-3. These results indicate that R_{max} decreases for m≥5. The reason is that the symbol erasure rate increases rapidly as a function of m. Undoubtedly, one would have to operate at a rate considerably less than R_{max}. to achieve a moderate gain. Thus, this approach is not feasible for obtaining high code rates at the larger values of m. However, for comparison purposes we did obtain a rough estimate of performance for codes with 5 and 6 bit symbols.

A conservative lower bound on performance can be obtained by using (D-2) to compute decoded bit error rate with symbol erasure rate given by (D-3) and assuming that symbol error rate is zero. Obviously, this technique will be far too optimistic at low values of E_N/N_o since the probability of symbol error will get rather large. Thus, this bound will give misleading results at lower code rates. However, the technique does show that the concatenated approach does not offer the possibility of achieving much coding gain at the higher code rates for δ=0.10. At R≈2/3 the concatenation scheme with a (31,25) RS code requires a symbol erasure rate of less than 0.03 to achieve $P_b=10^{-5}$. For a (63,49) RS code, a symbol erasure rate of 0.10 or less is required. Thus, from Figure D-2 $P_b=10^{-5}$ cannot be achieved at R≈2/3 with either m=5 or 6 with δ=0.10. At R≈1/2 one could use a concatenation scheme with a (31,19) RS code. This code requires a symbol erasure rate of 0.125 or less for $P_b=10^{-5}$. This is achieved at E_N/N_o=8.2 dB which corresponds to E_b/N_o=11.2 dB (for a 1.6 dB loss). A (63,37) RS code requires a symbol erasure rate of 0.21 or less. This is achieved at E_N/N_o=5.6 dB (E_b/N_o=8.6 dB) and thus shows the possibility of up to 1.0 dB gain. However, one must remember that we have used a lower bound on performance which gives an upper bound on gain. Use of this bound to evaluate performance at lower code rates would give misleading results. It is likely that this concatenated code approach would perform very well at code rates less than 1/2. However, since the technique does not perform well at higher code rates it is not of interest in our application.

Table D-3.

Maximum Rates for 1-dB Gain at δ=0.10

m	R_{max}
5	0.72
6	0.71
7	0.68
8	0.65
9	0.61
10	0.57


MED
8